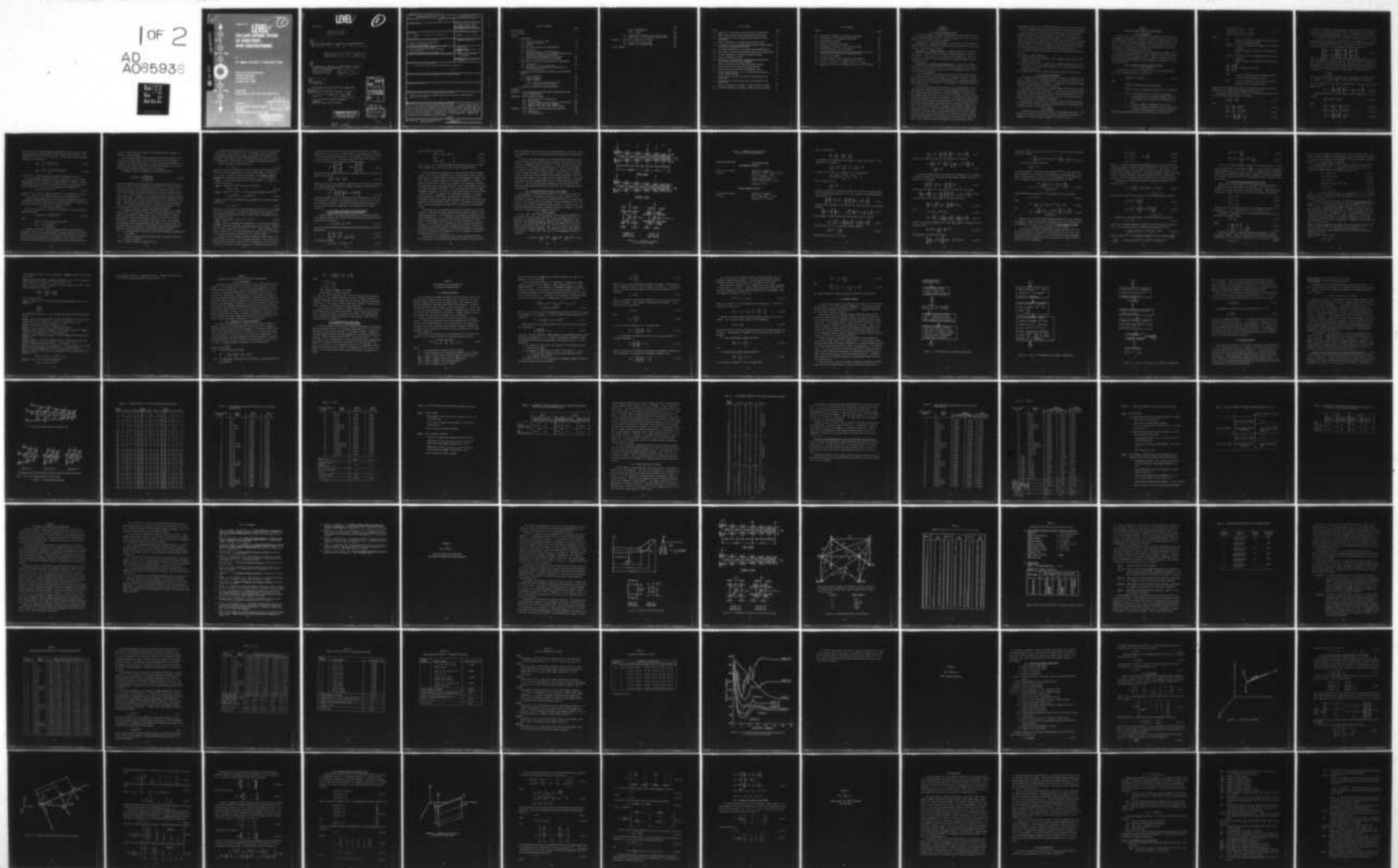
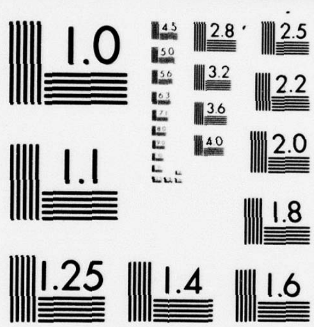
AD-A065 936 IOWA UNIV IOWA CITY DIV OF MATERIALS ENGINEERING F/G 1/3
FAIL-SAFE OPTIMAL DESIGN OF STRUCTURES WITH SUBSTRUCTURING.(U)
AUG 78 D T NGUYEN, A K GOVIL, J S ARORA DAAK11-77-C-0023

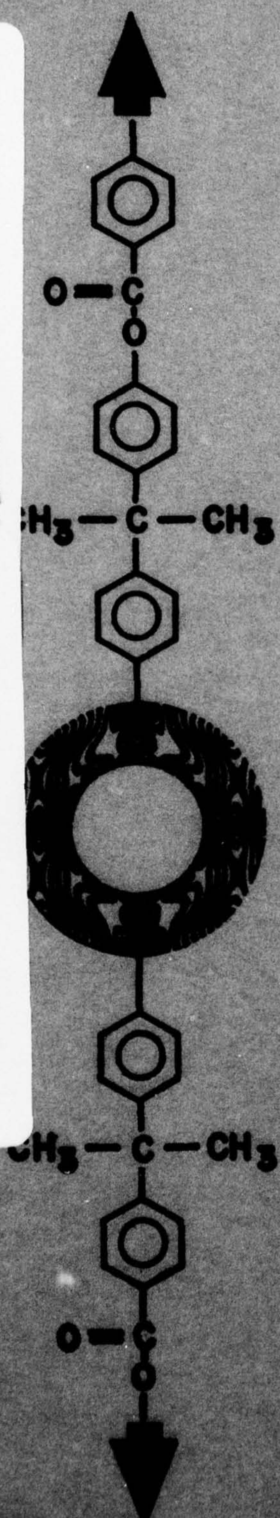
UNCLASSIFIED TR-45 NL

1 OF 2
AD
A065936





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



Report No. 45

LEVEL II

FAIL-SAFE OPTIMAL DESIGN OF STRUCTURES WITH SUBSTRUCTURING

by

D. T. Nguyen, A. K. Govil, J. S. Arora and E. J. Haug

Division of Materials Engineering
College of Engineering
The University of Iowa
Iowa City, Iowa 52242

August 1978
Interim Report for Period May 1977—August 1978

Prepared for
U.S. Army Ballistic Research Laboratory
DRDAR-BLB
Aberdeen Proving Ground, MD 21005

LEVEL II

1

Report No. 45

14 TR-45

6

FAIL-SAFE OPTIMAL DESIGN OF STRUCTURES WITH SUBSTRUCTURING.

by

15 DAAK11-77-C-0023

10

D. T. Nguyen, A. K. Govil, J. S. Arora ~~and~~ E. J. Haug
Division of Materials Engineering
College of Engineering
The University of Iowa
Iowa City, IA. 52242

12 188 p.

11

Aug~~ust 19~~78

Interim Report for Period May 1977 - August 1978

9 Interim technical rept. May 77- Aug 78,

Prepared for

U. S. Army Ballistic Research Laboratory
DRDAR-BLB
Aberdeen Proving Ground, MD 21005

79 03 19 090

409 250

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

5. TYPE OF REPORT & PERIOD COVERED
Interim Technical Report
May 1977 to August 1978

6. PERFORMING ORG. REPORT NUMBER
45

7. AUTHOR(s)
D. T. Nguyen, A. K. Govil, J. S. Arora and E. J. Haug

8. CONTRACT OR GRANT NUMBER(s)
DAAK11 - 77 - C - 0023

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Division of Materials Engineering
College of Engineering
The University of Iowa, Iowa City, Ia. 52242

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Ballistics Research Laboratory
ATTN: DRDAR-BLB
Aberdeen Proving Ground, MD 21005

12. REPORT DATE
August 1978

13. NUMBER OF PAGES
185

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Optimal Design; Fail-Safe; Damage Conditions; Structures; Substructuring; Helicopter Tail-Boom

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The problem of fail-safe structural design is defined. An optimal design formulation for the problem is developed with substructuring. An optimal design algorithm based on the state space gradient projection method is derived. Optimal designs for several cases of an open and closed tail-boom structures for the Army Cobra helicopter are obtained and analyzed. Constraints are imposed on stresses, displacements, buckling, natural frequency and member sizes. A user's manual for a computer program based on the algorithm is also described.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Purpose and Scope of Study

This report presents a systematic design approach that accounts for projected structural damage that may be inflicted during the life of a structure. This is called "fail-safe structural design."

Definition 1.1. Fail-Safe Structure: A structure is called fail-safe or damage tolerant if it continues to perform its basic functions even after sustaining a specified level of damage.

Definition 1.2. Damage Condition: A damage condition for a structure is defined as complete or partial removal of selected members or parts of the structure. Some joints of the structure may be removed as a result of the damage. A structure that has sustained the specified damage is called a damaged structure.

Definition 1.3. Optimal Fail-Safe Structure: A fail-safe or damage tolerant structure is called optimal if its design minimizes a cost function and satisfies constraints that must hold for the undamaged structure and for projected damage conditions.

A basic assumption in the method is that the structure remains geometrically stable after the specified damage to its members or joints. In other words the structure does not fail catastrophically in a mechanism-type motion after damage occurs. The structure is thus assumed to have enough redundancy in its construction.

One of the contributions of this report is in the development of a design sensitivity analysis method for fail-safe design with substructuring. Once design sensitivity information is known the designer can either use it in an optimal design procedure or he may use it to aid his intuition in adjusting design parameters to meet his objectives. Incorporation of substructuring in the fail-safe optimal design procedure is of critical importance since it makes the design sensitivity analysis and the structural analysis efficient. This allows the designer to consider a large number of damage conditions that may occur in large practical structures, without excessive computing effort. The main reason for this high efficiency is that when damage occurs to certain parts of the structure, the structural stiffness and mass matrices are modified only for those portions of the structure. This represents a small change in structural analysis with substructuring, whereas

without substructuring, the stiffness and mass matrices for the complete structure will be changed, making the structural analysis computationally expensive.

The optimal design algorithm for fail-safe structural design using the substructuring concept is first presented. The method is then applied to aircraft structures, such as a truss representation of the helicopter tail boom that was previously optimized by a similar method without substructuring. Optimum designs without substructuring that are obtained by using the computer code of Ref. 1 are given in Appendix A. Results obtained with the substructuring formulation are then compared with the previous results.

The optimal design algorithm takes into account the following considerations:

(a) Multiple loading conditions

(b) Various type of finite elements: truss, constant strain triangle, and symmetric shear panel

(c) Several elements of the structure may be assigned same design value and if required, can be kept fixed throughout or for a few iterations of the optimization process

(d) Damage that may occur to some elements and/or nodes of the structure.

1.2. Review of Literature

The concept of fail-safe optimal design of structures is relatively new. In Ref. 2 (Chapter 11), a comprehensive review of literature relative to fail-safe design of structures was conducted. No significant literature was found related to optimal design of fail-safe structures.

The concept of substructuring in optimal design of structures was recently presented by Govil, Arora and Haug [3]. It was shown that the idea of partitioning a large structure into a number of smaller substructures is profitable, since the total computational effort is reduced with incorporation of substructuring into the optimization algorithm.

The purpose of this report is to integrate concepts of fail-safe design and substructuring in order to develop and demonstrate an efficient approach to optimal design of fail-safe structures.

1.3. Notation

A standard matrix and vector notation is used throughout the report. All symbols are presumed to be matrices or vectors, unless stated otherwise. A superscript T is used to denote transpose of a matrix or vector.

CHAPTER 2

FAIL-SAFE OPTIMAL DESIGN WITH SUBSTRUCTURING

2.1. Introduction

In this chapter, the fail-safe optimal design problem with substructuring (FSODPS) is formulated. Constraints are imposed on member stresses, nodal displacements, and natural frequency under all loading and damage conditions. Constraints that are independent of load and damage conditions are also imposed. Design sensitivity analysis is developed and an algorithm is presented in a convenient step-by-step format.

The concepts of fail-safe design and substructuring in optimal structural design are presented in Refs. 2 and 3. Details of structural analysis with substructuring are presented in Ref. 4. However, structural analysis equations are required throughout the development of the algorithm, so they are summarized here.

2.2. Structural Analysis by Substructuring

2.2.1. Static Analysis

The equilibrium equation (state equation) in terms of displacements, for a given damaged condition α, is given as [4]:

$$K^{(\alpha)}(b)\, z^{(\alpha)} = S^{(\alpha)}(b) \tag{2.2-1}$$

where

$K^{(\alpha)}(b)$ = NxN structural stiffness matrix

$S^{(\alpha)}(b)$ = vector of N effective nodal loads on the structure

$z^{(\alpha)}$ = state variable vector of N nodal displacements

α = a superscript used to represent a damaged condition; for convenience $\alpha=0$ represents the undamaged structure.

N = number of degrees of freedom of the structure

b = a vector of D design variables, such as cross-sectional areas, moments of inertia, thickness and widths.

Using the substructuring concept, state equation 2.2-1 is written as:

$$\begin{bmatrix} K_{BB}^{(\alpha)} & K_{BI}^{(\alpha)} \\ K_{IB}^{(\alpha)} & K_{II}^{(\alpha)} \end{bmatrix} \begin{bmatrix} z_B^{(\alpha)} \\ z_I^{(\alpha)} \end{bmatrix} = \begin{bmatrix} S_B^{(\alpha)} \\ S_I^{(\alpha)} \end{bmatrix} \qquad (2.2\text{-}1)$$

where

B, I = subscripts refering to boundary and interior quantities for all substructures

$z_B^{(\alpha)} \varepsilon R^n$ = a vector of boundary displacements for the entire structure

n = boundary degrees of freedom for the entire structure

$z_I^{(\alpha)} \varepsilon R^m$ = a vector of interior displacements for the entire structure

m = interior degrees of freedom for the entire structure

$\left.\begin{matrix} K_{BB}^{(\alpha)}, K_{BI}^{(\alpha)} \\ K_{IB}^{(\alpha)}, K_{II}^{(\alpha)} \end{matrix}\right\}$ = submatrices of $K^{(\alpha)}(b)$

$S_B^{(\alpha)}$ = a vector of externally applied loads associated with the boundary degrees of freedom

$S_I^{(\alpha)}$ = a vector of externally applied loads associated with the interior degrees of freedom

Submatrices such ad $K_{BB}^{(\alpha)}$, $K_{BI}^{(\alpha)}$, $S_B^{(\alpha)}$, have compatible dimensions and will be understood to be functions of the design variable vector b.

The interior displacements $z_I^{(\alpha)}$ are first eliminated from Equation 2.2-2 and the following reduced equation is obtained

$$K_B^{(\alpha)} z_B^{(\alpha)} = F_B^{(\alpha)} \qquad (2.2\text{-}3)$$

where

$$K_B^{(\alpha)} = K_{BB}^{(\alpha)} + K_{BI}^{(\alpha)} Q^{(\alpha)} \qquad (2.2\text{-}4)$$

$$F_B^{(\alpha)} = S_B^{(\alpha)} + Q^{(\alpha)^T} S_I^{(\alpha)} \qquad (2.2\text{-}5)$$

$$Q^{(\alpha)} = - \left[K_{II}^{(\alpha)} \right]^{-1} K_{IB}^{(\alpha)} \qquad (2.2\text{-}6)$$

Here, $K_B^{(\alpha)}$ is a boundary stiffness matrix for the entire structure and $F_B^{(\alpha)}$ R^n is the vector of effective boundary forces. Efficient numerical procedures are used to decompose $K_{II}^{(\alpha)}$ and then to solve for $Q^{(\alpha)}$(mxn) in Equation 2.2-6.

The boundary stiffness $K_B^{(\alpha)}$ and the effective boundary force vector $F_B^{(\alpha)}$ are synthesized by considering contributions from all substructures. For this purpose, the equilibrium equation for a substructure, which is considered as an isolated free-body, is also expressed in the partitioned form

$$\begin{bmatrix} K_{BB}^{(r,\alpha)} & K^{(r,\alpha)} \\ K_{IB}^{(r,\alpha)} & K_{II}^{(r,\alpha)} \end{bmatrix} \begin{bmatrix} z^{(r,\alpha)} \\ z_I^{(r,\alpha)} \end{bmatrix} = \begin{bmatrix} S_B^{(r,\alpha)} \\ S_I^{(r,\alpha)} \end{bmatrix} \qquad (2.2\text{-}7)$$

where the superscript r refers to the r^{th} substructure and subscripts B and I refer to boundary and interior quantities. The vector $S_B^{(r,\alpha)}$ represents loads that are applied at the boundary nodes and reaction forces due to adjoining substructures. Let N(r) and m(r) represent the number of boundary and interior coordinates of the r^{th} substructure, respectively. It may be noted that

$$m = \sum_{r=1}^{L} m(r)$$

where L is the total number of substructures. Dimensions of various matrices are: $K_{BB}^{(r,\alpha)}$ is (n(r)x n(r)), $K_{IB}^{(r,\alpha)}$ is (m(r)x n(r)), $K_{BI}^{(r,\alpha)}$ is (n(r)x m(r))

$z_B^{(r,\alpha)}$ and $S_B^{(r,\alpha)} \epsilon R^{n(r)}$, and $z_I^{(r,\alpha)}$ and $S_I^{(r,\alpha)}$ $R^{m(r)}$. From the second line of Equation 2.2-7,

$$z_I^{(r,\alpha)} = \left[K_{II}^{(r,\alpha)}\right]^{-1} \left[S_I^{(r,\alpha)} - K_{IB}^{(r,\alpha)} \; z_B^{(r,\alpha)}\right] \qquad (2.2\text{-}8)$$

Substituting Equation 2.2-8 into the first line of Equation 2.2-7, one obtains:

$$K_B^{(r,\alpha)} \; z_B^{(r,\alpha)} = F_B^{(r,\alpha)} \qquad (2.2\text{-}9)$$

where

$$K_B^{(r,\alpha)} = K_{BB}^{(r,\alpha)} + K_{BI}^{(r,\alpha)} \; Q^{(r,\alpha)} \qquad (2.2\text{-}10)$$

$$F_B^{(r,\alpha)} = S_B^{(r,\alpha)} + Q^{(r,\alpha)^T} \; S_I^{(r,\alpha)} \qquad (2.2\text{-}11)$$

$$Q^{(r,\alpha)} = -\left[K_{II}^{(r,\alpha)}\right]^{-1} \; K_{IB}^{(r,\alpha)} \qquad (2.2\text{-}12)$$

The $(n(r) \times n(r))$ boundary stiffness matrix $K_B^{(r,\alpha)}$ and the $(n(r) \times 1)$ effective boundary force vector $F_B^{(r,\alpha)}$ for each substructure are computed from Equations 2.2-10 and 2.2-11, respectively. Finally, K_B and F_B are assembled according to the equations

$$K_B^{(\alpha)} = \sum_{r=1}^{L} \beta^{(r)^T} K_B^{(r,\alpha)} \beta^{(r)} \qquad (2.2\text{-}13)$$

$$F_B^{(\alpha)} = S_B^{(\alpha)} + \sum_{r=1}^{L} \beta^{(r)^T} Q^{(r,\alpha)^T} S_I^{(r,\alpha)} \qquad (2.2\text{-}14)$$

where $\beta^{(r)}$ is a Boolean transformation matrix of dimension $(n(r) \times n)$.

Using the reduced equilibrium equation of Equation 2.2-3, the boundary displacements $z_B^{(\alpha)}$ are computed by a suitable numerical procedure. Interior displacements are then computed for each substructure, using Equation 2.2-8. Lastly, member-end forces for the r^{th} substructure are computed from

$$p^{(r,\alpha)} = K^{(r,\alpha)} z^{(r,\alpha)} \qquad (2.2\text{-}15)$$

where $p^{(r,\alpha)}$ is a vector of member forces, $K^{(r,\alpha)}$ is a stiffness matrix and $z^{(r,\alpha)}$ is a vector of nodal displacements for the r^{th} substructure.

Multiple loading conditions for the structure are treated routinely by taking $S^{(\alpha)}(b)$ and $z^{(\alpha)}$ in Equation 2.2-1 as matrices whose j^{th} columns represent quantities associated with the j^{th} loading condition.

2.2.2 Frequency Analysis

The natural frequency of a structure is computed by solving the general eigenvalue proglem

$$K^{(\alpha)}(b)y^{(\alpha)} = \zeta^{(\alpha)} M^{(\alpha)}(b)y^{(\alpha)} \qquad (2.2\text{-}16)$$

where

$M^{(\alpha)}(b)$ = (NxN) structural mass matrix

$y^{(\alpha)}$ = an eigenvector

$\zeta^{(\alpha)}$ = an eigenvalue

A number of techniques, such as Subspace Iteration [5], Householder's method [6], a method based on Sturm sequence properties described by Gupta [7], Wilkinson [8], and others [9,10] are available in the literature for solution of the general eigen-problem defined in Equation 2.2-16. However, these techniques require computation and decomposition of stiffness and mass matri-

ces for the entire structure, which is not desirable since it defeats the purpose of substructuring.

There are many component mode substitution techniques available in the literature that may be used. For a complete survey of such techniques, the reader is referred to Reference 11. These techniques take advantage of substructuring. However, they are not suitable for integration into an optimum design algorithm, becuase they are not efficient.

A technique, based on minimization of the Rayleigh Quotient

$$R^{(\alpha)}(y^{(\alpha)}) = \frac{y^{(\alpha)^T} K^{(\alpha)} y^{(\alpha)}}{y^{(\alpha)^T} M^{(\alpha)} y^{(\alpha)}} \qquad (2.2\text{-}17)$$

has been discussed and used successfully by researchers such as Fox and Kapoor [12], Wilkinson [8], and Bradbury and Fletcher [13]. This method does not require storage of the matrices K and M for the entire structure, because all calculations can proceed elementwise to obtain a solution of Equation 2.2-17. However, there is one difficulty with this procedure of computing eigenvalues. Convergence to an eigenvalue and the corresponding eigenvector can be quite slow if a good initial estimate of the eigenvector is not known. Some methods of selecting initial eigenvectors have been suggested [12,13], but no general procedure exists to alleviate this problem. Therefore this method is also not suitable for general applications.

The Subspace Iteration technique [5] generally converges to an eigensolution in only a few iterations. The method converges quite rapidly eventhough a poor estimate of eigenvectors is used. This technique, however, also requires calculation and storage of matrices K and M for the entire structure [5]. Therefore, the method in its present form is not suitable for integration into the optimal design algorithm with substructuring. However the method can be modified for incorporation into the substructuring algorithm. This new approach has the following desirable features:

(i) It converges rapidly even when a good initial estimate of eigenvectors is not known

(ii) It does not require calculation and storage of matrices K and M for the entire structure

(iii) It does not require decomposition of K.

The method of Subspace Iteration can be used to solve any desired number of eigenvalues of the Equation 2.2-16. The Subspace Iteration algorithm is first summarized without partitioning the structure into a number of substructures. Then modifications to the algorithm are presented that account for partitioning of the structure into a number of smaller substructures.

Consider the general eigenvalue problem

$$K \, \Phi = M \, \Phi \, \Omega \tag{2.2-18}$$

where K and M are the stiffness and the mass matrices for the structure, Φ is an $(N \times p)$ matrix of eigenvectors, p is the desired number of eigenvalues, and Ω is a $(p \times p)$ diagonal matrix of eigenvalues. The <u>Subspace Iteration algorithm</u> for computing p eigenvalues of Equation 2.2-18 is as follows:

<u>Step 1</u>. Start with $(N \times q)$ matrix $X^{(0)}$ as an estimate of q eigenvectors; $q = \min \{2p, p+8, N\}$.

<u>Step 2</u>. Compute $Y^{(0)} = MX^{(0)}$, and solve for $\overline{X}^{(1)}$ from

$$K\overline{X}^{(1)} = Y^{(0)} \tag{2.2-19}$$

<u>Step 3</u>. Compute $\overline{Y}^{(1)} = M\overline{X}^{(1)}$. Calculate the following $(q \times q)$ matrices

$$\overline{K} = \overline{X}^{(1)^T} Y^{(0)} , \qquad \overline{M} = \overline{X}^{(1)^T} Y^{(1)} \tag{2.2-20}$$

<u>Step 4</u>. Solve for all eigenvalues and eigenvectors of the reduced eigenvalue problem

$$\overline{K} \, \overline{\Phi} = \overline{M} \, \overline{\Phi} \, \overline{\Omega} \tag{2.2-21}$$

where $\overline{\Phi}$ is a $(q \times q)$ matrix of reduced eigenvectors and $\overline{\Omega}$ is a $(q \times q)$ diagonal matrix of eigenvalues. Note that the generalized Jacobi iteration or the determinant search method [5] may be used to solve the eigenvalue problem of Equation 2.2-21.

<u>Step 5</u>. Compute $X^{(1)} = \overline{X}^{(1)} \, \overline{\Phi}, \; Y^{(1)} = \overline{Y}^{(1)} \, \overline{\Phi}$ (2.2-22)

<u>Step 6</u>. Check for convergence of eigenvalues. If all eigenvalue changes are within a specified tolerance, then stop the iterative process. Otherwise return to Step 1 with $X^{(0)} = X^{(1)}$ and $Y^{(0)} = Y^{(1)}$. After convergence, the first p columns of $X^{(1)}$ are required eigenvectors and the first p eigenvalues in $\overline{\Omega}$ are the corresponding eigenvalues of the original system.

In order to use the Subspace Iteration method with substructuring, one needs to modify only Step 2 of the preceding algorithm. If one can use the substructuring procedure to solve for $\overline{X}^{(1)}$ from Equation 2.2-19, then he has a method for efficiently solving the structural eigenvalue problem by par-

titioning the structure into a number of smaller substructures. Comparing Equations 2.2-1 and 2.2-19, one observes that the two equations are similar, so the substructuring approach used to solve Equation 2.2-1 can also be used to solve Equation 2.2-19. Accordingly, matrices $X^{(1)}$ and $Y^{(0)}$ in Equation 2.2-19 are partitioned into boundary and interior parts as

$$X^{(1)} = \begin{bmatrix} X_B^{(1)} \\ X_I^{(1)} \end{bmatrix}, \quad Y^{(0)} = \begin{bmatrix} Y_B^{(0)} \\ Y_I^{(0)} \end{bmatrix} \tag{2.2-23}$$

Following the same approach as for static structural analysis, one solves for $X_B^{(1)}$ from the equation

$$K_B X_B^{(1)} = Y_B^{(0)} + Q^T Y_I^{(0)} \tag{2.2-24}$$

where matrices K_B and Q are defined in Equations 2.2-4 and 2.2-6, respectively. The interior displacements $X_I^{(1)}$ are computed, as before, substructure-wise. For the r^{th} substructure

$$\overline{X}_I^{r(1)} = \left[K_{II}^{(r)}\right]^{-1} \left[Y_I^{r(0)} + Q^{(r)} \overline{X}_B^{r(1)}\right] \tag{2.2-25}$$

where matrices $K_{II}^{(r)}$ and $Q^{(r)}$ are defined earlier in this section. Note that the superscript α is omitted from Equations 2.2-23 to 2.2-25. This is done for notational convenience. The modified Subspace Iteration algorithm is used to calculate natural frequencies of the undamaged and all damaged structures.

2.3. State Space Definition of Fail-Safe Optimal Design Problem with Substructuring (FSODPS)

A general FSODPS in the state space setting may be defined as follows: Find a design variable vector b that, under both complete and damaged states, minimizes a cost function

$$J = J\left(b, z_B^{(\alpha)}, z_I^{(\alpha)}, \zeta^{(\alpha)}\right) \tag{2.3-1}$$

satisfies the partitioned equilibrium equitions (state equations) in terms of displacements

$$K_B^{(\alpha)} z_B^{(\alpha)} = F_B^{(\alpha)} \tag{2.3-2}$$

$$K_{II}^{(\alpha)} z_I^{(\alpha)} = S_I^{(\alpha)} - K_{IB}^{(\alpha)} z_B^{(\alpha)} \tag{2.3-3}$$

the eigenvalue problem

$$K^{(\alpha)} y^{(\alpha)} = \zeta^{(\alpha)} M^{(\alpha)} y^{(\alpha)} \tag{2.3-4}$$

and satisfies the constraints

$$\phi^{s(\alpha)}\ (b,\ z_B^{(\alpha)},\ z_I^{(\alpha)}) \leq 0 \tag{2.3-5}$$

$$\phi^{d}(b) \leq 0 \tag{2.3-6}$$

$$\phi^{e(\alpha)}\ (\zeta^{(\alpha)}) \leq 0 \tag{2.3-7}$$

for $\alpha = 0,1,2,\ldots,\bar{d}$. Here $F_B^{(\alpha)}$ and $K_B^{(\alpha)}$ are defined in Equations 2.2-5 and 2.2-4, respectively, and $\bar{d}$ is the total number of damage conditions

The cost function of Equation 2.3-1 is quite general and may represent weight of the structure, displacements of critical points, certain critical member forces, or perhaps natural frequency of the undamaged or damaged structure. The cost function depends only on design variables if it represents weight of the structure. The vector inequality 2.3-5 represents constraints that depend upon state and design variables. These are the member stress and the nodal displacement constraints. It is noted here that some constraints represented in Equation 2.3-5 will not depend explicitly on all the parameters b, $z_B^{(\alpha)}$, and $z_I^{(\alpha)}$. For example, the displacement constraint at boundary nodes depends only on $z_B^{(\alpha)}$ and at interior nodes it depends only on $z_I^{(\alpha)}$. For members connected to boundary and interior nodes, stress constraints will depend on all the parameters b, $z_B^{(\alpha)}$, and $z_I^{(\alpha)}$. Advantages of these special forms of various constraint functions will be realized in all calculations [3,14].

The inequality 2.3-6 represents constraints that depend only on design variables. These include either explicit bounds on design variables or relationship between them. The inequality 2.3-8 represents a constraint on the lowest eigenvalue ($\zeta \geq \zeta_0$; ζ_0 = allowable lowest eigenvalue) which may be related to the fundamental frequency of the structure (frequency $f = \sqrt{\zeta}/2\pi$ Hertz). In the present work, constraints on only the lowest eigenvalue are considered, but constraints on higher eigenvalues can also be included [2]. The lowest eigenvalue of Equation 2.3-4 may also be related to the buckling load for the structure [4] and Equation 2.3-7 will represent a constraint on the buckling load for the structure.

The FSODPS is now formulated in terms of state and design variables. This formulation of the optimal design problem is superior to purely design

space formulation since it permits one to take advantage of the form of structural equations to carry out the design sensitivity analysis very efficiently [15,16].

In order to show the advantage of the substructuring formulation over a similar formulation without substructuring, consider the Helicopter Tail Boom structure of Appendix A (shown in Figure 2.1). With the present formulation, the structure is divided into three substructures (r=1,2,3) by partitioning it at nodes 9-12 and 17-20 as shown in Figure 2.1. Suppose that the damage occurs in members belonging to Substructure 2. Table 2.1 shows a comparison of major calculations with and without substructuring formulations. With the substructuring formulation one always works with smaller matrices. Also, substructures that have no damaged members do not require calculation of $K_{II}^{(r,\alpha)}$ and $Q^{(r,\alpha)}$. Thus the substructuring formulation of the fail-safe optimal structural design problem should be more efficient than a formulation without substructuring.

2.4. Design Sensitivity Analysis of the FSODPS

The philosophy of the optimization method is to start with the best engineering estimate of the design variable vector b and to improve it until an optimum is reached. Thus, one must determine the effect of a design change on the cost and constraint functions before a design improvement δb can be calculated. This is known as deisgn sensitivity analysis. In this section, first the design sensitivity analysis of a general function is considered. Then the analysis is specialized to the FSODPS of Section 2.3.

2.4.1 General Approach

Let $\psi(b, z_B^{(\alpha)}, z_I^{(\alpha)}, \zeta^{(\alpha)})$ be a general function that may represent the cost function or any constraint function for the αth damage condition ($\alpha = 0$ represents the undamaged structure). When the design is changed by a small amount δb, the displacements $z_B^{(\alpha)}$ and $z_I^{(\alpha)}$ and the eigenvalue $\zeta^{(\alpha)}$ will also change by small amounts $\delta z_B^{(\alpha)}$, $\delta z_I^{(\alpha)}$, and $\delta\zeta^{(\alpha)}$ due to the well posed nature of the structural analysis problem. Let $\delta\zeta$ represent a first order change in the function ψ. Taking b, $z_B^{(\alpha)}$, $z_I^{(\alpha)}$, and $\zeta^{(\alpha)}$ as independent variables, $\delta\psi$ is given as

$$\delta\psi = \frac{\partial\psi}{\partial b}\,\delta b + \frac{\partial\psi}{\partial z_B^{(\alpha)}}\,\delta z_B^{(\alpha)} + \frac{\partial\psi}{\partial z_I^{(\alpha)}}\,\delta z_I^{(\alpha)} + \frac{\partial\psi}{\partial\zeta^{(\alpha)}}\,\partial\zeta^{(\alpha)} \qquad (2.4\text{-}1)$$

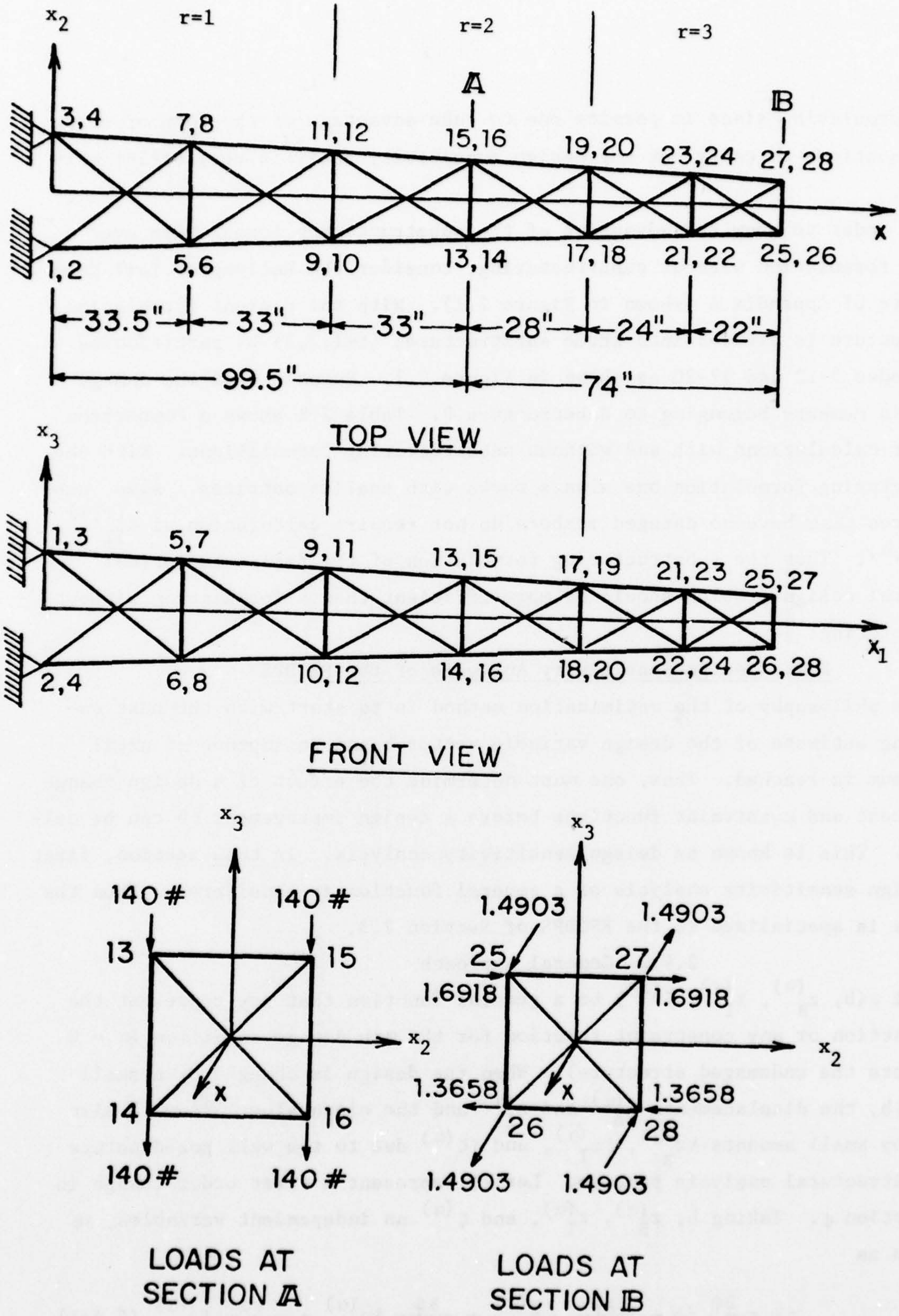


Figure 2.1. Arrangement of Members for Open Truss Tail-Boom

TABLE 2.1. COMPARISON OF CALCULATIONS
WITH AND WITHOUT SUBSTRUCTURING

Without substructuring	With substructuring
For undamaged structure	
Generate and decompose K (72,21)	Generate and decompose $K_B^{(0)}(36,12)$, $K_{II}^{(1,0)}$, $K_{II}^{(2,0)}$, $K_{II}^{(3,0)}$: each of dimension (12,12). Generate from Equation 2.2-12 $Q^{(1,0)}$, $Q^{(2,0)}$, $Q^{(3,0)}$: each of dimension (12,24)
For each damaged condition	
Generate and decompose K (72,21)	Generate and decompose $K_B^{(\alpha)}$ (36,12), $K_{II}^{(2,\alpha)}$ (12,12) Calculate $Q^{(2,\alpha)}$ (12,24)

where the derivatives

$$\frac{\partial\psi}{\partial b}, \frac{\partial\psi}{\partial z_B^{(\alpha)}}, \frac{\partial\psi}{\partial z_I^{(\alpha)}}, \frac{\partial\psi}{\partial\zeta^{(\alpha)}}$$

are computed at the previously known values of b, $z_B^{(\alpha)}$, $z_I^{(\alpha)}$ and $\zeta^{(\alpha)}$. The problem is now to express

$$\frac{\partial\psi}{\partial z_B^{(\alpha)}} \partial z_B^{(\alpha)}, \frac{\partial\psi}{\partial z_I^{(\alpha)}} \partial z_I^{(\alpha)}, \frac{\partial\psi}{\delta\zeta^{(\alpha)}} \delta\zeta^{(\alpha)}$$

in terms of δb, so that $\delta\psi$ in Equation 2.4-1 is expressed as

$$\frac{\partial\psi(b, z_B^{(\alpha)}(b), z_I^{(\alpha)}(b), \zeta^{(\alpha)}(b))}{\partial b} \delta b$$

First consider the term

$$\frac{\partial\psi}{\partial z_I^{(\alpha)}} \delta z_I^{(\alpha)}.$$

In order to obtain this expression in terms of δb, define the following identity by premultiplying Equation 2.3-3 by the transpose of an adjoint variable vector $\lambda_I^{(\alpha)}$(m x n):

$$\left[\lambda_I^{(\alpha)}\right]^T K_{II}^{(\alpha)} z_I^{(\alpha)} = \left[\lambda_I^{(\alpha)}\right]^T \left[S_I^{(\alpha)} - K_{IB}^{(\alpha)} z_B^{(\alpha)}\right] \tag{2.4-2}$$

Taking the first variation of this identy in b, $z_B^{(\alpha)}$, and $z_I^{(\alpha)}$, one rearranges to obtain

$$\left[K_{II}^{(\alpha)} \lambda_1^{(\alpha)}\right]^T \left[\delta z_I^{(\alpha)} = \lambda_I^{(\alpha)}\right]^T \left[C_2^{(\alpha)} \delta b - K_{IB}^{(\alpha)} \delta z_B^{(\alpha)}\right] \tag{2.4-3}$$

where symmetry of $K_{II}^{(\alpha)}$ has been used and the matrix $C_2^{(\alpha)}$ is given as

$$C_2^{(\alpha)} = \frac{\partial S_I^{(\alpha)}}{\partial b} - \frac{\partial}{\partial b}\left[K_{IB}^{(\alpha)} z_B^{(\alpha)}\right] - \frac{\partial}{\partial b}\left[K_{II}^{(\alpha)} z_I^{(\alpha)}\right] \tag{2.4-4}$$

If one now selects $\lambda_I^{(\alpha)}$ to satisfy the adjoint equation

$$K_{II}^{(\alpha)} \lambda_I^{(\alpha)} = \frac{\partial\psi^T}{\partial z_I^{(\alpha)}} \tag{2.4-5}$$

then Equations 2.4-3 and 2.4-5 yield

$$\frac{\partial\psi}{\partial z_I^{(\alpha)}}\ \delta z_I^{(\alpha)} = \left[\lambda_I^{(\alpha)}\right]^T \left[C_2^{(\alpha)}\delta b - K_{IB}^{(\alpha)}\ \delta z_B^{(\alpha)}\right] \tag{2.4-6}$$

Equation 2.4-6 may be substituted into Equation 2.4-1 to obtain

$$\delta\psi = \left[\frac{\partial\psi}{\partial b} + \lambda_I^{(\alpha)^T}\ C_2^{(\alpha)}\right]\delta b + \left[\frac{\partial\psi}{\partial z_B^{(\alpha)}} - \lambda_I^{(\alpha)^T}\ K_{IB}^{(\alpha)}\right]\partial\, z_B^{(\alpha)} + \frac{\partial\psi}{\partial\zeta^{(\alpha)}}\,\delta\zeta^{(\alpha)} \tag{2.4-7}$$

To obtain an expression for the second term of Equation 2.4-7 in terms of δb, define an identity by introducing another adjoint variable vector $\lambda_B^{(\alpha)}$ in Equation 2.3-2 as follows:

$$\left[\lambda_B^{(\alpha)}\right]^T\ K_B^{(\alpha)}\ z_B^{(\alpha)} = \left[\lambda_B^{(\alpha)}\right]^T\ F_B^{(\alpha)} \tag{2.4-8}$$

Taking the first variation of this identity in b and $z_B^{(\alpha)}$, one obtains

$$\left[K_B^{(\alpha)}\ \lambda_B^{(\alpha)}\right]^T\ \delta z_B^{(\alpha)} = \left[\lambda_B^{(\alpha)}\right]^T\left[\frac{\partial F_B^{(\alpha)}}{\partial b} - \frac{\partial}{\partial b}\{K_B^{(\alpha)}\ z_B^{(\alpha)}\}\right]\delta b \tag{2.4-9}$$

It can be shown [3] that the identity 2.4-9 may be written as

$$\left[K_B^{(\alpha)}\ \lambda_B^{(\alpha)}\right]^T\ \delta z_B^{(\alpha)} = \left[\lambda_B^{(\alpha)}\right]^T\ C^{(\alpha)}\delta b \tag{2.4-10}$$

where

$$C^{(\alpha)} = C_1^{(\alpha)} + Q^{(\alpha)^T}\ C_2^{(\alpha)} \tag{2.4-11}$$

$$C_1^{(\alpha)} = \frac{\partial S_B^{(\alpha)}}{\partial b} - \frac{\partial}{\partial b}\left[K_{BB}^{(\alpha)}\ z_B^{(\alpha)}\right] - \frac{\partial}{\partial b}\left[K_{BI}^{(\alpha)}\ z_I^{(\alpha)}\right] \tag{2.4-12}$$

and $Q^{(\alpha)}$ is defined in Equation 2.2-6. Now, select $\lambda_B^{(\alpha)}$ to be solution of the adjoint equation

$$K_B^{(\alpha)}\ \lambda_B^{(\alpha)} = \frac{\partial\psi^T}{\partial z_B^{(\alpha)}} - K_{BI}^{(\alpha)}\ \lambda_I^{(\alpha)} \tag{2.4-13}$$

Then Equations 2.4-10 and 2.4-13 yield

$$\left[\frac{\partial\psi}{\partial z_B^{(\alpha)}} - \lambda_I^{(\alpha)^T}\ K_{IB}^{(\alpha)}\right]\delta z_B^{(\alpha)} = \lambda_B^{(\alpha)^T}\ C^{(\alpha)}\ \delta b \tag{2.4-14}$$

where $K_{BI}^{(\alpha)} = K_{IB}^{(\alpha)}$ has been used. Equation 2.4-14 is now substituted into Equation 2.4-7 to obtain

$$\delta\psi = \left[\frac{\partial\psi}{\partial b} + \lambda_I^{(\alpha)^T} C_2^{(\alpha)} + \lambda_B^{(\alpha)^T} C^{(\alpha)}\right]\delta b + \frac{\partial\psi}{\partial\zeta^{(\alpha)}}\,\partial\zeta^{(\alpha)} \qquad (2.4\text{-}15)$$

Now one must treat the expression

$$\frac{\partial\psi}{\partial\zeta^{(\alpha)}}\,\delta\zeta^{(\alpha)} \; .$$

The design sensitivity analysis of the eigenvalue $\zeta^{(\alpha)}$ has been considered by many researchers [8]. Therefore, this development is only summarized. From the first order expansion of Equation 2.3-4 and using the fact that $K^{(\alpha)}$ and $M^{(\alpha)}$ are symmetric, one obtains the following expression for $\delta\zeta^{(\alpha)}$:

$$\delta\zeta^{(\alpha)} = \frac{y^{(\alpha)^T}\dfrac{\partial}{\partial b}\left[K^{(\alpha)}y^{(\alpha)} - \zeta^{(\alpha)}M^{(\alpha)}y^{(\alpha)}\right]\delta b}{y^{(\alpha)^T}M^{(\alpha)}y^{(\alpha)}} \; . \qquad (2.4\text{-}16)$$

Substituting this expression for $\delta\zeta^{(\alpha)}$ in Equation 2.4-15, one obtains

$$\delta\psi = G^{(\alpha)^T}\,\delta b \qquad (2.4\text{-}17)$$

where

$$G^{(\alpha)} = \frac{\partial\psi}{\partial b}^T + C_2^{(\alpha)^T}\lambda_I^{(\alpha)} + C^{(\alpha)^T}\lambda_B^{(\alpha)} + \Lambda^{e(\alpha)} \qquad (2.4\text{-}18)$$

and

$$\Lambda^{e(\alpha)} = \frac{\dfrac{\partial}{\partial b}\left[K^{(\alpha)}y^{(\alpha)} - \zeta^{(\alpha)}M^{(\alpha)}y^{(\alpha)}\right]^T y^{(\alpha)}\,\dfrac{\partial\psi}{\partial\zeta^{(\alpha)}}}{y^{(\alpha)^T}M^{(\alpha)}y^{(\alpha)}} \; . \qquad (2.4\text{-}19)$$

Equation 2.4-17 is the desired relationship between the design change and the change in a member force, a nodal displacement, the cost function, and/or an eigenvalue. The vector $G^{(\alpha)}$ is the required <u>design sensitivity vector</u>.

2.4.2. Design Sensitivity Matrices for the FSODPS

In the fail-safe optimal design problem, design sensitivity vectors of active constraints are calculated for one damage condition at a time. Once the design sensitivity analysis of all active constraints under all damage conditions has been completed, then first variations of constraint Equations 2.3-5 to 2.3-7 are expressed as

$$\delta \tilde{\phi}^{s} = \Lambda^{s^T} \delta b \tag{2.4-20}$$

$$\delta \tilde{\phi}^{s} = \Lambda^{d^T} \delta b \; ; \qquad \Lambda^{d} = \frac{\partial \tilde{\phi}^{d^T}}{\partial b} \tag{2.4-21}$$

$$\delta \tilde{\phi}^{e} = \Lambda^{e^T} \delta b \tag{2.4-22}$$

where a '~' over a constraint function represents inclusion of only active or violated constraints. Matrices Λ^s and Λ^d have D rows. The number of columns in each depends upon the total number of violations in s and d types of constraints for all damage conditions. Each column represents a design sensitivity vector. Similarly, the matrix Λ^e stores sensitivity vectors obtained from Equation 2.4-19 for violated frequency constraints for all damage conditions.

The matrix Λ^s can be easily obtained by following the approach of the previous section:

$$\Lambda^{s(\alpha)} = \frac{\partial \tilde{\phi}^{s(\alpha)^T}}{\partial b} + C_2^{(\alpha)^T} \lambda_I^{s(\alpha)} + C^{(\alpha)^T} \lambda_B^{s(\alpha)} \tag{2.4-23}$$

$$\alpha = 0,1,\ldots\bar{d}$$

Matrices $\lambda_I^{s(\alpha)}$ and $\lambda_B^{s(\alpha)}$ are solutions of the following adjoint equations:

$$K_{II}^{(\alpha)} \lambda_I^{s(\alpha)} = \frac{\partial \tilde{\phi}^{s(\alpha)^T}}{\partial z_I^{(\alpha)}} \tag{2.4-24}$$

$$K_B^{(\alpha)} \lambda_B^{s(\alpha)} = \frac{\partial \tilde{\phi}^{s(\alpha)^T}}{\partial z_I^{(\alpha)}} + Q^{(\alpha)^T} \frac{\partial \tilde{\phi}^{s(\alpha)^T}}{\partial z_I^{(\alpha)}} \tag{2.4-25}$$

$$\alpha = 0,1,\ldots\bar{d}$$

Similarly a first order change in the cost function is expressed as

$$\delta J = \Lambda^{J^T} \delta b \tag{2.4-26}$$

where Λ^J is the design sensitivity vector for the cost function of Equation 2.3-1. This sensitivity vector is obtained from Equation 2.4-18 as

$$\Lambda^{J} = \frac{\partial J^T}{\partial b} + C_2^{(\alpha)^T} \lambda_I^{J(\alpha)} + C^{(\alpha)^T} \lambda_B^{J(\alpha)} + \Lambda^{J(\alpha)} \tag{2.4-27}$$

Here, the vector $\Lambda^{J(\alpha)}$ is obtained from Equation 2.4-19 by replacing $\frac{\partial \psi}{\partial \zeta^{(\alpha)}}$ by $\frac{\partial J}{\partial \zeta^{(\alpha)}}$. Adjoint vectors $\lambda_I^{J(\alpha)}$ and $\lambda_B^{J(\alpha)}$ are solutions of

$$K_{II}^{(\alpha)} \lambda_I^{J(\alpha)} = \frac{\partial J^T}{\partial z_I^{(\alpha)}} \tag{2.4-28}$$

$$K_B^{(\alpha)} \lambda_B^{J(\alpha)} = \frac{\partial J^T}{\partial z_B^{(\alpha)}} + Q^{(\alpha)} \frac{\partial J}{\partial z_I^{(\alpha)}} \tag{2.4-29}$$

If the cost function represents weight of the structure, then J is a function of b only and Λ^J is simply given as $\frac{\partial J^T}{\partial b}$. If the cost function depends on other variables such as z_B, z_I, and ζ for the α^{th} damage condition (for example one may want to maximize the lowest natural frequency or minimize displacement at some point of the structure), then the sensitivity vector is given by Equation 2.4-27.

2.5. Optimal Design Algorithm for the FSODPS

Restrictions are now placed on the linearized constraint functions. It is required that the design change δb be computed in such a manner that it corrects, or at least improves, all violated constraints. These requirements on Equations 2.4-20 to 2.4-22 can be stated as the following inequalities:

$$\Lambda^{s^T} \delta b \leq \Delta\tilde{\phi}^s \tag{2.5-1}$$

$$\Lambda^{d^T} \delta b \leq \Delta\tilde{\phi}^d \tag{2.5-2}$$

$$\Lambda^{e^T} \delta b \leq \Delta\tilde{\phi}^e \tag{2.5-3}$$

where $\Delta\tilde{\phi}^s$, $\Delta\tilde{\phi}^d$, $\Delta\tilde{\phi}^e$ are desired corrections in constraint violations. If a constraint $\phi_i < 0$ is ε- active (that is, $\phi_i \geq -\varepsilon$), then $\Delta\tilde{\phi}_i = -\phi_i$.

Constraints of Equations 2.5-1 to 2.5-3 have similar forms, so they can be written in a compact form as

$$\Lambda^T \delta b \leq \Delta\tilde{\phi} \tag{2.5-4}$$

where

$$\Lambda = \left[\Lambda^s \ \Lambda^d \ \Lambda^e \right] \tag{2.5-5}$$

$$\Delta\tilde{\phi} = \left[\Delta\tilde{\phi}^{s^T} \ \ \Delta\tilde{\phi}^{d^T} \ \ \Delta\tilde{\phi}^{e^T} \right]^T \tag{2.5-6}$$

The reduced problem of computing an optimum design change δb can now be stated as follows: Find δb to minimize the cost function of Equations 2.4-26 subject to the constraint of Equation 2.5-4 and a step size constraint

$$\delta b^T W \delta b < \xi^2 \tag{2.5-7}$$

where W is a positive definite weighting matrix and ξ is a small number. The matrix W (usually diagonal) is used to assign weights to the various components of δb and is often essential when components of b represent different physical quantities of different orders of magnitude.

The reduced FSODPS defined in the preceeding is exactly the problem defined in References 14 and 17. An application of Kuhn-Tucker conditions of nonlinear programming gives the following solution [14,17]:

$$\delta b = -\eta \delta b^1 + \delta b^2 \tag{2.5-8}$$

$$\delta b^1 = W^{-1} [\Lambda^J + \Lambda \mu^1] \tag{2.5-9}$$

$$\delta b^2 = -W^{-1} \Lambda \mu^2 \tag{2.5-10}$$

$$H[\mu^1 \vdots \mu^2] = [(-\Lambda^T W^{-1} \Lambda^J) \vdots -\Delta\tilde{\phi}] \tag{2.5-11}$$

$$H = \Lambda^T W^{-1} \Lambda \tag{2.5-12}$$

$$\mu = \mu^1 + (1/\eta) \mu^2 \tag{2.5-13}$$

where $\eta > 0$ is a step size to be chosen by the designer and $\mu \geq 0$ is a Lagrange multiplier vector. The method of step size selection is the same as used in References 1-3, 14, 17.

The method can now be described by the following step-by-step algorithm:

Step 1. At the j^{th} design point $b^{(j)}$ and under α^{th} damaged condition (if $\alpha > \bar{d}$, go to Step 11), generate matrices $K_{II}^{(r,\alpha)}$ and $K_{IB}^{(r,\alpha)}$ for each substructure. Note superscript r denotes r^{th} substructure. Decompose each $K_{II}^{(r,\alpha)}$ and calculate the matrix $Q^{(r,\alpha)}$ from Equation 2.2-12. Store decomposed part of the matrix $K_{II}^{(r,\alpha)}$ and the matrix $Q^{(r,\alpha)}$ for later calculations. Calculate the boundary stiffness matrix $K_B^{(r)}$ and the effective boundary load vector $F_B^{(r)}$ from Equations 2.2-13 and 2.2-14, respectively. Decompose the matrix $K_B^{(r)}$ and store it for later use.

Step 2. Calculate boundary displacements $z_B^{(r)}$ from Equation 2.3-2 and interior displacements $z_I^{(r,\alpha)}$ for each substructure from Equation 2.2-8.

Step 3. Calculate the lowest eigenvalue and the corresponding eigenvector from Equation 2.2-16.

Step 4. Compute adjoint vectors

$$\lambda_I^{J(\alpha)}, \quad \lambda_B^{J(\alpha)}$$

from Equations 2.4-28 to 2.4-29, respectively. Assemble the matrix Λ^J of Equation 2.4-27.

Step 5. Check the frequency constraint of Equation 2.3-7. If it is violated, then compute $\Lambda^{e(\alpha)}$ of Equation 2.4-19 and put $\Delta\tilde{\phi}^e = -\tilde{\phi}^e_i$.

Step 6. Check constraints of Equations 2.3-5 and form the vector $\tilde{\phi}^{s(\alpha)}$. Calculate the sensitivity information

$$\frac{\partial\tilde{\phi}^{s(\alpha)}}{\partial b}, \frac{\partial\tilde{\phi}^{s(\alpha)}}{\partial z_B^{(\alpha)}}, \frac{\partial\tilde{\phi}^{s(\alpha)}}{\partial z_I^{(\alpha)}}$$

Also, calculate $\Delta\tilde{\phi}^{s(\alpha)}$.

Step 7. Calculate $\lambda_I^{s(r,\alpha)}$ for each substructure from Equation 2.4-24. Note $K_{II}^{(r,\alpha)}$ and

$$\frac{\partial\tilde{\phi}^{s(\alpha)}}{\partial z_I^{(r,\alpha)}}$$

are completely uncoupled [3]. Also, calculate the matrix $\lambda_B^{s(\alpha)}$ from Equation 2.4-25.

Step 8. Calculate the matrices $C_1^{(\alpha)}$ and $C_2^{(\alpha)}$ from Equations 2.4-12 and 2.4-14, respectively. Also, calculate the matrix $C^{(\alpha)}$ from Equation 2.4-11.

Step 9. Assemble the matrix Λ^s of Equation 2.4-23.

Step 10. If $\alpha \geq \bar{d}$, go to Step 11, otherwise go to Step 1.

Step 11. Check constraints of Equation 2.3-7 and form a vector $\tilde{\phi}^d$. Compute the matrix Λ^d of Equation 2.4-21. Also, compute $\Delta\tilde{\phi}^d$.

Step 12. Finally, assemble the matrix Λ and $\Delta\tilde{\phi}$ of Equations 2.5-5 and 2.5-6, respectively.

Step 13. Compute μ^1 and μ^2 from Equation 2.5-11. Choose a step size η and compute the Lagrange multiplier vector μ from Equation 2.5-13.

Step 14. Check the sign of each component of μ. If any component of μ is negative, remove corresponding rows from Λ^T and $\Delta\tilde{\phi}$ and return to Step 13.

Step 15. Compute δb^1, δb^2, and δb from Equations 2.5-9, 2.5-10 and 2.5-8, respectively. Let

$$b^{(j+1)} = b^{(j)} + \delta b , \; y^{(j+1)} = y^{(j)}.$$

Step 16. If all constraints are satisfied and

$$||\delta b^1|| = [\delta b^{1^T} W \delta b^1]^{1/2}$$

is sufficiently small [17], terminate the process. Otherwise, return to Step 1 with $b^{(j+1)}$ as the best available design, and set $\alpha = 0$.

CHAPTER 3

DISCUSSION OF THE METHOD AND COMPUTATIONAL CONSIDERATIONS

3.1. Introduction

The method for fail-safe optimal structural design with substructuring of Chapter 2 is quite general since no assumption is made regarding the type of finite elements to be employed. However the algorithm presented in Section 2.5 requires considerable computation for even moderate size structures. It is noted that computational techniques such as design variable linking, the ϵ-active constraint concept and normalization of constraints with respect to their limit values are easily incorporated in the present algorithm [1,3,14]. Further, all computational considerations in structural analysis, design sensitivity analysis and Lagrange multiplier calculations used in optimal design of structures with substructuring [3,18,19] are also incorporated in the algorithm for the FSODPS. Finally, for step size determination, convergence criterion, and computational checks, the reader is referred to References 2 and 3. In this chapter, only those computational aspects of the method that are different from those presented in References 1 to 3 are discussed.

3.2. Selection of Critical Constraints

The FSODPS is characterized by requiring a set of design variables to satisfy a constraint set whose dimension is much larger (due to damage and multiple loading conditions) than the dimension of the design variable vector. If all active constraints come into the computation, it is not only computationally expensive but the accuracy of the result may be jeopardized. Thus a rational method of selecting independent critical constraints is essential. In the present work, the idea of "worst violated constraint" [3,14] is used in order to eliminate redundent constraints. The i^{th} constraint is

$$\phi_i \leq 0$$

where ϕ_i is defined as

(i) for the i^{th} stress constraint

$$\phi_i^{(\alpha)} = \max_{\alpha,j,k} \left\{ \phi_{ijk}^{s(\alpha)}(b, z_B^{(\alpha)}, z_I^{(\alpha)}) \right\}$$

(ii) for displacement constraint of the i^{th} degree of freedom and α^{th} damage condition

$$\phi_i^{(\alpha)} = \max_k \left\{ \phi_{ik}^{s(\alpha)} \left(z_B^{(\alpha)}, z_I^{(\alpha)} \right) \right\}$$

Here

$\alpha = 0,1,2,\ldots,\bar{d}$

$j = 1,2,\ldots,NMG$

$k = 1,2,\ldots,NLC$

NMG = number of members in the group

NLC = number of loading conditions.

Thus, for stress constraints only the worst violation over all members of a group, over all loading conditions, and over all damage conditions, is imposed. Similarly, for any damage condition the worst violated displacement constraint at a node over all loading conditions is imposed. The natural frequency constraint is imposed for all damage conditions.

In this procedure, the number of violated constraints to be corrected at any design iteration is reduced considerably. The procedure avoids calculation of design derivatives of unnecessary constraints. The Lagrange multiplier calculations of these constraints are also avoided. Thus, efficiency of the algorithm is enhanced.

3.3. Some Additional Computational Condiderations in Structural Analysis

In static analysis of structures, the response variables to be determined under each damage conditon (α) are boundary displacements $z_B^{(\alpha)}$, interior displacements for each substructure $z_I^{(r,\alpha)}$, and element stresses. Computation of these response quantities requires generation of the boundary stiffness matrix $K_B^{(\alpha)}$, interior stiffness matrices $K_I^{(r,\alpha)}$, and matrices $Q^{(r,\alpha)}$. It is noted that in case no damage occurs in some substructures under a damage condition, then for those substructures computation of the matrices $K_{II}^{(r,\alpha)}$ and $Q^{(r,\alpha)}$ is not required. This increases efficiency of the algorithm, since as generation and decomposition of $K_{II}^{(r,\alpha)}$ and computation of $Q^{(r,\alpha)}$ are not required. The remaining computations proceed as discussed in References 3, 18 and 19.

CHAPTER 4

APPLICATION OF THE ALGORITHM FOR FAIL-SAFE STRUCTURAL DESIGN

4.1 Design Formulation

In this chapter, the general method for fail-safe optimal structural design developed in Chapters 2 and 3 is specialized to structures that can be modeled by TRUSS, Constant Strain Triangular (CST), and/or Symmetric Shear Panel (SSP)/Symmetric Pure Shear Panel (SPSP) finite elements. Stiffness and mass matrices for these elements are given in Appendix B. For the class of structures considered herein, the geometric configuration and the material for the structure are assumed to be specified. Loading conditions and probable damage conditions for the structure are also specified.

An optimal design problem for this class of structures is defined as follows: find the cross-sectional area of TRUSS elements and the thickness of the CST and SSP/SPSP elements so that total weight of the structure is minimized and the state equations and constraints on stress, buckling, displacement, natural frequency, and member size are satisfied for all loading and damage conditions.

Since weight of the structure is to be minimized, the cost function of Equation 2.3=1 is a linear function of the design variables, given as

$$J(b) = \sum_{r=1}^{L} \sum_{k=1}^{TP(r)} \sum_{i=1}^{NG(k)} \sum_{j=1}^{NM(i)} \rho_i \ell_{ij} b_i \qquad (4.1\text{-}1)$$

where:

ρ_i = material density of members in the i^{th} group,

b_i = cross-sectional area or thickness of members in the i^{th} group,

ℓ_{ij} = length or surface area of the j^{th} member in the i^{th} group,

$TR(r)$ = number of element types in the r^{th} substructure,

$NG(r)$ = number of groups in the r^{th} substructure,

$NM(i)$ = number of members in the i^{th} group.

Since the cost function depends only on design variables, the vector Λ^J of Equation 2.4-27 is simply $\frac{\partial J^T}{\partial b}$.

In the following presentation, the superscript α designating a damage condition is omitted in all equations. These equations apply to a typical damage condition. In this formulation, CST/SSP/SPSP elements are required to satisfy a design criterion based on the Von Mises equivalent stress. For a complete development of the Von Mises equivalent stress criterion, the reader is referred to Ref. 20. According to this criterion, an equivalent stress (σ^c) for a structural element in a general state of stress is given as

$$\sigma^c = \left[\frac{1}{2}\{(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2 + 6(\sigma_{12}^2+\sigma_{23}^2+\sigma_{31}^2)\}\right]^{1/2} \tag{4.1-2}$$

where σ_{ij} $(i,j = 1,2,3)$ are stress components at the point of interest (x_1,x_2,x_3) in the domain Ω of the element. For CST or SSP/SPSP elements, Equation 4.1-2 reduces to

$$\sigma^c = (\sigma_{11}^2+\sigma_{22}^2-\sigma_{11}\,\sigma_{22}+\,3\sigma_{12}^2)^{1/2} \tag{4.1-3}$$

Next, the stress or buckling constraint of Section 3.2 for a typical member is written as

$$\phi_i^s = \left|\frac{\sigma_i^c}{\sigma^a}\right| - 1.0 \leq 0 \tag{4.1-4}$$

where σ_i^c is the direct stress for TRUSS elements, or the maximum Von Mises stress calculated from Equation 4.1-3. In order to simultaneously implement stress and buckling constraints for truss members, the allowable stress σ^a is chosen as follows:

(i) for members in tension, $\sigma^a = \sigma^{a+}$, where σ^{a+} is an allowable tensile stress for the member

(ii) for members in compression, $\sigma^a = \min(\sigma^{a-}, \sigma^b)$, where $\sigma^{a-} > 0$ and $\sigma^b > 0$ are allowable compressive and critical buckling stresses for the member, respectively.

The stresses σ^{a+} and σ^{a-} are specified by the designer, whereas σ^b depends on the Euler buckling load and is given as

$$\sigma_i^b = \frac{\pi^2 E I_i}{\ell_i^2 b_i} \tag{4.1-5}$$

where E, ℓ_i, b_i, and I_i are modulus of elasticity, length, cross-sectional area, and moment of inertia of the i^{th} member, respectively. In the present work, it is assumed that the moment of inertia of a truss member can be expressed as

$$I_i = \bar{\alpha}_i b_i^2 \tag{4.1-6}$$

where $\bar{\alpha}_i$ is a positive constant that depends only on the cross-sectional geometry of the member. This constant is specified by the designer. Thus, Equation 4.1-5 is rewritten as

$$\sigma_i^b = \bar{\theta}_i b_i \tag{4.1-7}$$

where

$$\bar{\theta}_i = \frac{\pi^2 E \bar{\alpha}_i}{\ell_i^2} \tag{4.1-8}$$

If the constraint of Equation 4.2-4 is violated, then

$$\Delta\tilde{\phi}^s = - \left[\left| \frac{\sigma^c}{\sigma^a} \right| - 1.0 \right] \tag{4.1-9}$$

The displacement constraint of Section 3.2 for a typical degree of freedom is expressed as

$$\phi_j^s \equiv \left| \frac{z_j}{z_j^a} \right| - 1.0 \leq 0 \tag{4.1-10}$$

where z_j and z_j^a are the calculated and allowable displacements, respectively. If this constraint is violated for a displacement component, then

$$\Delta\tilde{\phi}_j^s = - \left[\left| \frac{z_j}{z_j^a} \right| - 1.0 \right] \tag{4.1-11}$$

It is noted here that constraint checks on stress, buckling, and displacement proceed substructurewise. The sensitivity analysis proceeds as explained in Chapter 2 and the matrix Λ^s is assembled at this stage.

The constraint of Equation 2.5-3 is imposed only on the lowest eigenvalue of the structure ($\zeta = (2\pi f)^2$). Using the method presented in Section 2.2.2 the lowest eigenvalue ζ and the associated eigenvector y are obtained. Thus, the eigenvalue constraint is written as

$$\phi^e(\zeta) \equiv 1.0 - \zeta/\zeta_0 \leq 0 \qquad (4.1\text{-}12)$$

where ζ_0 is related to a resonant frequency of the structure. If this constraint is violated, then

$$\Delta\tilde{\phi}^e = -(1.0 - \zeta/\zeta_0) \quad \text{and} \quad \frac{\partial\tilde{\phi}^e}{\partial\zeta} = -\frac{1}{\zeta_0} \qquad (4.1\text{-}13)$$

Finally, the design variable constraint $\phi^d(b)$ of Equation 2.5-2 is considered. For a typical design variable it is expressed as

$$b_i^L \leq b_i \leq b_i^U \qquad (4.1.14)$$

where b_i^L and b_i^U are the lower and upper bounds on the i^{th} design variable, respectively. The inequality of Equation 4.1-14 may be split into two parts as follows:

(i) Lower bound design variable constraint

$$\phi_i^d(b) \equiv 1.0 - \frac{b_i}{b_i^L} \leq 0 \qquad (4.1\text{-}15)$$

and

(ii) Upper bound design variable constraint

$$\phi_i^d(b) \equiv \frac{b_i}{b_i^U} - 1.0 \leq 0 \qquad (4.1\text{-}16)$$

If a constraint of Equation 4.1-15 is violated, then

$$\Delta\tilde{\phi}_i^d = -\left(1.0 - \frac{b_i}{b_i^L}\right) \tag{4.1-17}$$

and

$$\frac{\partial\tilde{\phi}_i^d}{\partial b} = \left[\ 0,\ldots,\ 0,\ -\frac{1}{b_i^L}\ ,\ 0,\ldots,0\right] \tag{4.1-18}$$

The upper bound design variable constraint is treated in a similar way.

4.2 Computer Program

A computer program based on the formulation of Section 4.1 and the algorithm of Chapter 2, has been developed in FORTRAN IV. The program is called SOS4 (Structural Optimization by Substructures 4). A general flow diagram for the program is given in Figure 4.1.

Computational aspects of multiple loading conditions, design variable linking, the worst constraint violation concept, the ε-active constraint concept, and normalization of constraints have been incorporated in the program [3,4,18,19]. For frequency analysis of the structure (calculation of lowest natural frequency), estimates of two eigenvectors are needed. These eigenvector estimates are either supplied by the designer or are generated internally by the computer program at the start of the iterative design process. In all subsequent frequency analyses, eigenvectors from the previous frequency analysis are taken as the starting eigenvectors.

In order to obtain a reasonable starting design for the algorithm, a stress-ratio design is made in the computer code. In this concept, member areas for TRUSS elements and member thicknesses for CST and/or SSP/SPSP elements are computed from the condition that stress in each member be at its limiting value. It is noted here that this does not necessarily yield an optimum design even under only stress constraints for indeterminate structures, but it gives a good starting point for the optimal design algorithm. A parameter IFS is defined in the computer program for controlling the number of stress-ratio design cycles.

Also, provision is made in the computer program for assigning a predetermined value to any design variable at the start of the iterative process.

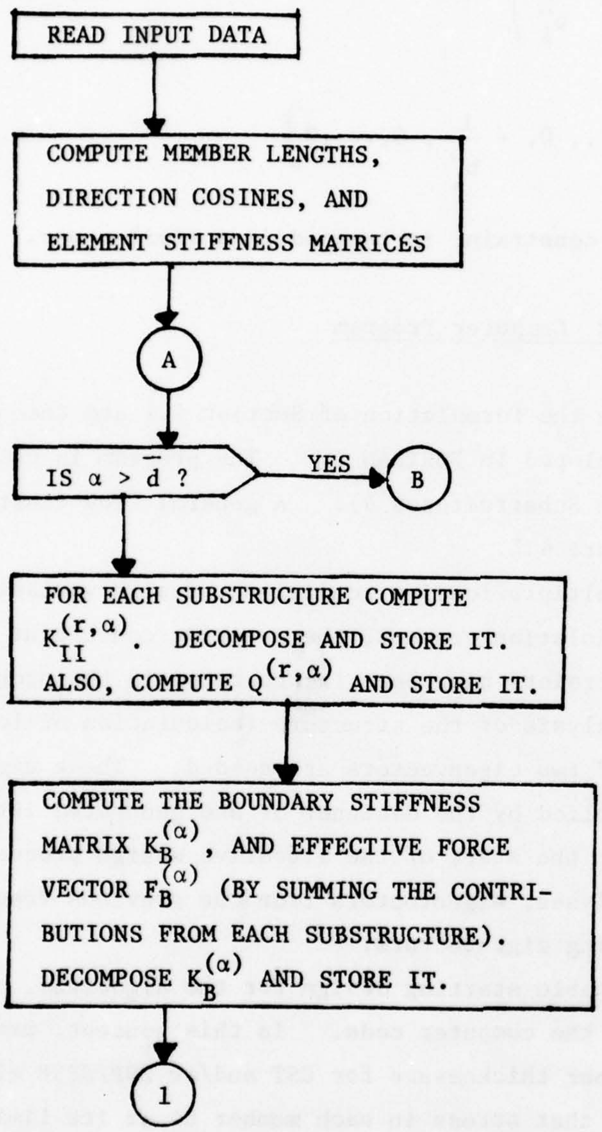


Figure 4.1. Flow Diagram for the Computer Program SOS4

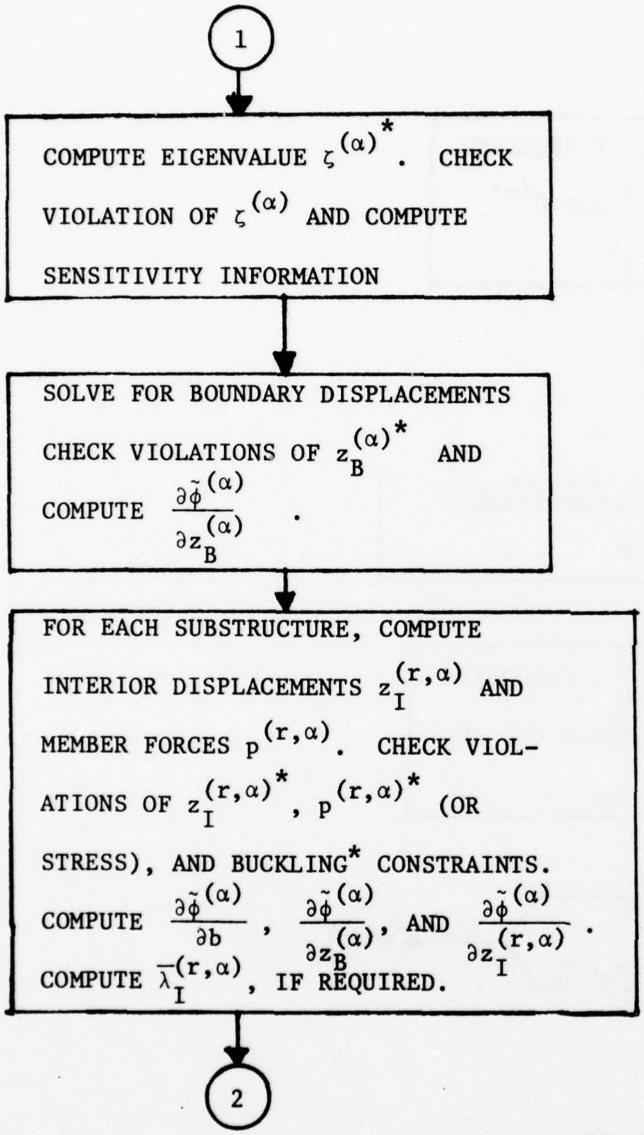


Figure 4.1. (cont.) Flow Diagram for the Computer Program SOS4.

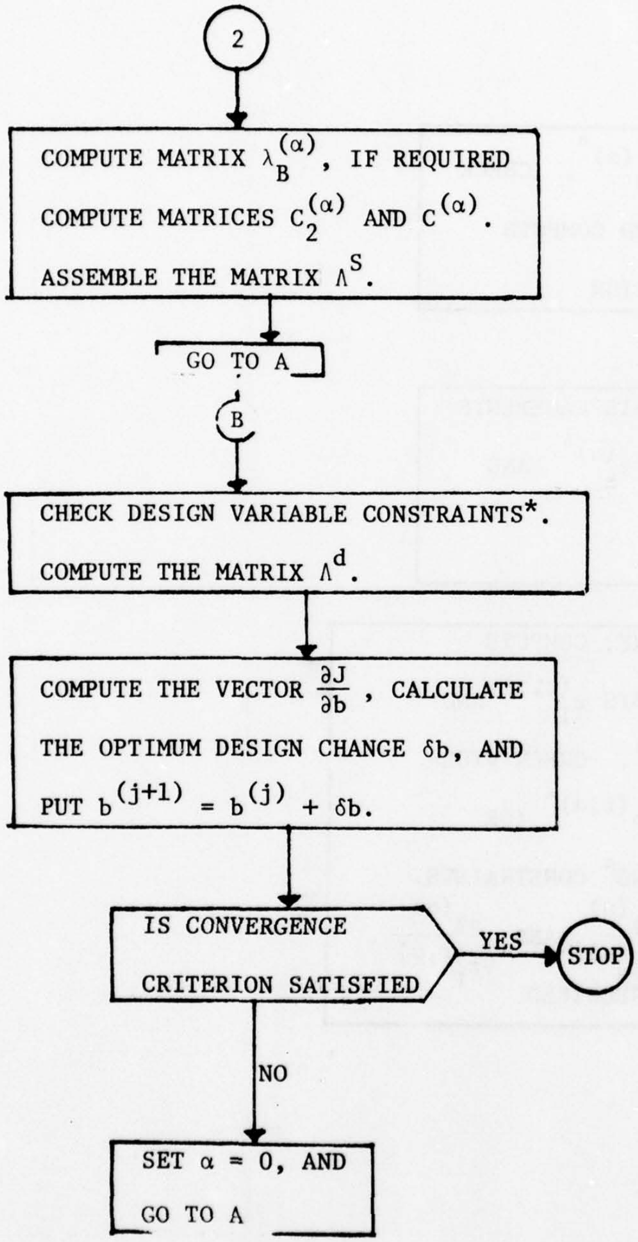


Figure 4.1. (cont.) Flow Diagram for the Computer Program SOS4.

Thus, the number of design variables can be less than the number of groups for the structure. This is a valuable feature in the program, since it allows a designer to fix some members of the structure. Stress constraints, however, are imposed on all members of the structure, irrespective of whether the design variable associated with the member is fixed or free.

The step size η is calculated in the program based on a desired reduction in the cost function when all constraints are satisfied [3,14]. Thus, if $\bar{r}$ is a desired cost function reduction ratio, then the step size is given as

$$\eta = \bar{r}J/\left(\Lambda^{J^T}\Lambda^J\right) \tag{4.2-1}$$

The weighting coefficients W_i are calculated, as in Ref. 3, as

$$W_i = \frac{\partial J}{\partial b_i}\bar{w}_i \tag{4.2-2}$$

where $\bar{w}_i$ is a weighting multiplier. *It is noted here that selection of the* parameters $\bar{r}$ and $\bar{w}_i$ is still an art. A certain amount of expreience is necessary to choose effective values for these parameters. Well chosen values of these parameters are necessary for rapid convergence of the algorithm. In many example problems $\bar{r}$ has been chosen as 0.05 to 0.10. The multiplier $\bar{w}_i$ has been chosen as unity for the CST and SSP/SPSP elements and for TRUSS elements it has been chosen between 1 and 20.

4.3 Example Problems

The formulation of Section 4.1 is now used to design a tail-boom structure for the U.S. Army Cobra helicopter. Geometry of the tail-boom structure and the loads transmitted to it are shown in Figure A.1. Fail-safe design for several cases of the tail-boom modeled as an open truss structure are given in Appendix A. Those designs were obtained using the computer code of Ref. 1 which is based on an optimal design formulation without substructuring. In this section the following two design problems are solved using the substructuring formulation and the computer program SOS4:

Design Problem 1: Open truss helicopter tail boom

Design Problem 2: Closed helicopter tail boom

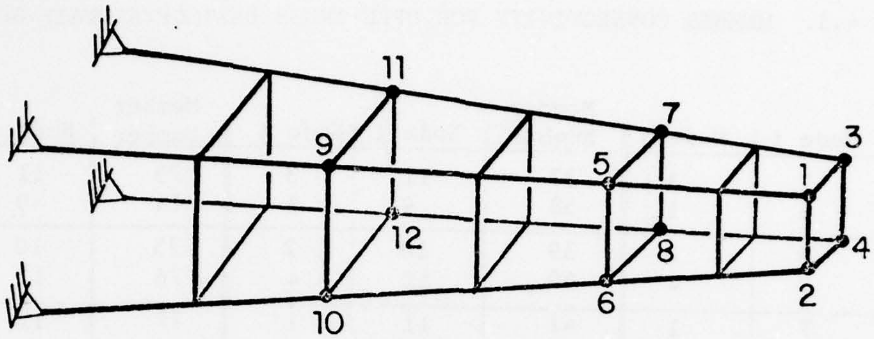
In subsequent subsections these problems are discussed in detail. Results obtained for the first example are compared to results given in Appendix A. Finally, results for the tail-boom obtained with and without substructuring are compared.

4.3.1 Open Truss Helicopter Tail-Boom

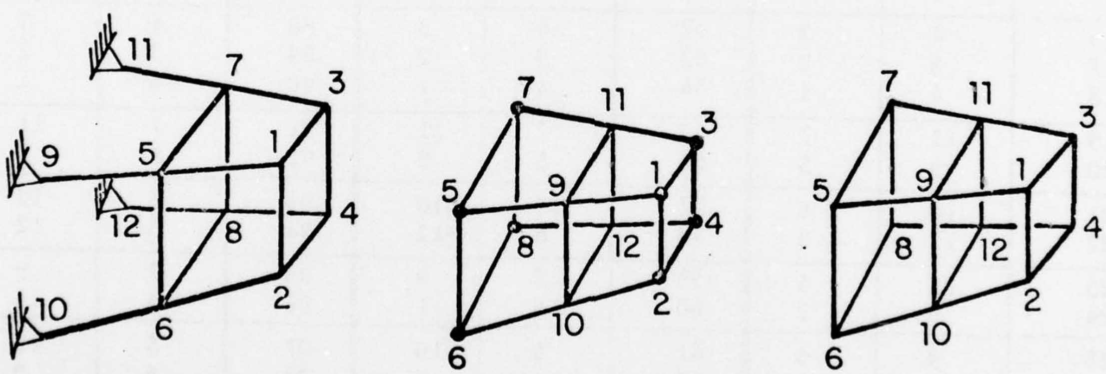
A complete description of the structure, such as member arrangement, global coordinate axes, loading data, definition of 6 damage conditions and other design data are given in Appendix A. In the present formulation, the tail-boom structure (Fig. 2.1) is divided into 3 substructures by partitioning it at nodes 9-12 and 17-20. Node numbers 25-28 are also treated as boundary nodes. Figure 4.2 shows nodal numbering and the local (or substructural) numbering systems. Substructure 1 has 4 boundary nodes (1-4), 8 interior nodes (5-12) and 36 truss elements. Substructures 2 and 3 each have 8 boundary nodes (1-8), 4 interior nodes (9-12), and 36 truss elements. Note that boundary nodes 1-4 of the first substructure and 1-8 of the second and third substructures correspond to boundary nodes 9-12, 5-12, and 1-8 in the overall numbering system, respectively. Member connectivity is given in Table 4.1. Design variable linking is used, as shown in Table 4.2.

In this example, a starting design of 1.0 in^2. for all members of the structure is used. Optimum designs for two cases are obtained using the computer program SOS4. These are Cases I and V of Appendix A. For Case I, no damage is considered and for Case V, six damage conditions are imposed. The final designs, cost function, number of active constraints, $||\delta b^1||$ at optimum, maximum $||\delta b^1||$, and average CPU time per iteration are given in Table 4.2. Critical constraints at the optimum are given in Table 4.3.

Comparision of Results Obtained with and without Substructuring: A comparison of optimum results obtained with and without substructuring formulations is given in Table 4.4. Optimum weights using the two formulations are the same. However, the CPU time per design iteration using the substructuring approach is increased by a factor of 1.4 for Case I and 1.6 for Case V. An analysis of



(a) An Overall Numbering System for Boundary Nodes



substructure 1 substructure 2 substructure 3

(b) Numbering System for Boundary and Interior Nodes for Each Substructure

Note: For clarity diagonal members are not shown.

Figure 4.2. Nodal Numbering Systems

TABLE 4.1. MEMBER CONNECTIVITY FOR OPEN TRUSS HELICOPTER TAIL-BOOM

Member Number	Node i	Node j	Member Number	Node i	Node j	Member Number	Node i	Node j
1	7	3	37	11	3	73	11	3
2	5	1	38	9	1	74	9	1
3	6	2	39	10	2	75	10	2
4	8	4	40	12	4	76	12	4
5	7	1	41	11	1	77	11	1
6	5	3	42	9	3	78	9	3
7	5	2	43	9	2	79	9	2
8	6	1	44	10	1	80	10	1
9	6	4	45	10	4	81	10	4
10	8	2	46	12	2	82	12	2
11	7	4	47	11	4	83	11	4
12	8	3	48	12	3	84	12	3
13	3	1	49	3	1	85	3	1
14	1	2	50	1	2	86	1	2
15	2	4	51	2	4	87	2	4
16	4	3	52	4	3	88	4	3
17	3	2	53	3	2	89	3	2
18	4	1	54	4	1	90	4	1
19	11	7	55	7	11	91	7	11
20	9	5	56	5	9	92	5	9
21	10	6	57	6	10	93	6	10
22	12	8	58	8	12	94	8	12
23	11	5	59	7	9	95	7	9
24	9	7	60	5	11	96	5	11
25	9	6	61	5	10	97	5	10
26	10	5	62	6	9	98	6	9
27	10	8	63	6	12	99	6	12
28	12	6	64	8	10	100	8	10
29	11	8	65	7	12	101	7	12
30	12	7	66	8	11	102	8	11
31	7	5	67	11	9	103	11	9
32	5	6	68	9	10	104	9	10
33	6	8	69	10	12	105	10	12
34	8	7	70	12	11	106	12	11
35	7	6	71	11	10	107	11	10
36	8	5	72	12	9	108	12	9

TABLE 4.2. FINAL DESIGN FOR OPEN TRUSS HELICOPTER TAIL-BOOM WITH SUBSTRUCTURING

Design Variable Number	Member Number	CASE I Area, in^2.	CASE V Area, in^2.
1	2,3	1.1825	2.2751
2	1,4	1.1804	2.1222
3	5,6,9,10	0.1705	0.3817
4	7,8,11,12	0.1874	0.4137
5	13,15	0.0415	0.0448
6	14,16	0.1572	0.1511
7	17,18	0.0415	0.1242
8	20.21	1.2931	3.0785
9	19,22	1.2936	2.8335
10	23,24,27,28	0.1359	0.2549
11	25,26,29,30	0.1707	0.2196
12	31,33	0.0415	0.1027
13	32,34	0.1266	0.1693
14	35,36	0.0415	0.3946
15	38,39	0.8076	0.9388
16	37,40	0.8076	0.9824
17	41,42,45,46	0.2440	0.3910
18	43,44,47,48	0.2763	0.1445
19	49,51	0.0415	0.0876
20	50,52	0.1864	0.1135
21	53,54	0.0415	0.1881
22	56,57	0.9938	1.2978
23	55,58	0.9922	1.2641
24	59,60,63,64	0.2141	0.3242
25	61,62,65,66	0.2581	0.2897
26	67,69	0.0415	0.0910

TABLE 4.2 Cont'd

Design Variable Number	Member Number	CASE I Area, in^2.	CASE V Area, in^2.
27	68,70	0.1835	0.0922
28	71,72	0.0415	0.0836
29	74,75	0.2322	0.2673
30	73,76,	0.2326	0.1231
31	77,78,81,82	0.3413	0.1938
32	79,80,83,84	0.3508	0.3262
33	85,87	0.0458	0.2115
34	86,88	0.1023	0.0814
35	89,90	0.2062	0.1806
36	92,93	0.5787	0.5571
37	91,94	0.5787	0.6898
38	95,96,99,100	0.2764	0.2872
39	97,98,101,102	0.3036	0.3108
40	103,105	0.0415	0.0442
41	104,106	0.2031	0.1508
42	107,108	0.0415	0.1155
Weight in lbs.		106.0	161.55
Average CPU/iter. in sec. (IBM 370-158)		5.65	43.50
No. of Active Constr. at opt.		12	9
$\|\|\delta b^1\|\|$ at opt.		4.37	6.9
$\|\|\delta b^1\|\|_{max}$		53.92	53.8

TABLE 4.3 CRITICAL CONSTRAINTS AT OPTIMUM (OPEN TRUSS TAIL-BOOM)

CASE I: Without damage

- Displacement in the x_2 direction at nodes 1 and 3 of the 3rd substructure
- Lower limit on design variable numbers 5,7,12,14,19,21, 26,28,40 and 42
- Max. violation is 0.001% at optimum

CASE V: With 6 damaged conditions

- Frequency constraint under damage conditions 2 and 6
- Displacement in the x_2 direction at node 1 of the 3rd substructure under damage conditions 2,3,4 and 5
- Displacement in the x_2 direction at node 3 of the 3rd substructure under damage conditions 2,3, and 5
- Max. violation is 0.09% at optimum

TABLE 4.4. COMPARISON OF RESULTS OBTAINED WITH AND WITHOUT SUBSTRUCTURING FOR OPEN TRUSS HELICOPTER TAIL-BOOM

	CASE I		CASE V		
	Optimum Weight	CPU Time per design iteration	Optimum Weight	CPU Time per design iteration	Computer Core Requirements
Without Substructuring	105.6	4.0	161.1	26.7	280 K
With Substructuring	106.0	5.65	161.7	43.5	276 K

computer programs SOS4 and that of Ref. 1 showed that there is some difference in the frequency analysis portion of the two programs. In the computer program of Ref. 1, the mass matrix for the structure is calculated and stored for use in Steps 2 and 3 of the Subspace Iteration method of Section 2.2.2. However, in the SOS4 computer program the mass matrix for the structure is not stored. Multiplication of the mass matrix with eigenvectors in Steps 2 and 3 of the Subspace Iteration method of Section 2.2.2 is carried out elementwise. Thus, if the Subspace Iteration method takes approximately 5 cycles to converge to the eigenvalue solution for a structure, then the mass matrix is computed 10 times in the program SOS4, as compared to only once in the program of Ref. 1. When six damage conditions are imposed, the mass matrix is calculated 70 times in the program SOS4, as compared to only 7 times in the program of Ref. 1, in each design iteration. Therefore, the frequency analysis portion of the program of Ref. 1 is more efficient as compared to that in the program SOS4. However, for the two approaches, there is a trade-off between computational time and computer storage.

In order to confirm the preceding contention, the two computer programs were executed without the frequency analysis for a case of the open truss helicopter tail-boom with six damage conditions imposed. The program SOS4 took 13.0 sec. per design iteration, whereas the program of Ref. 1 took 14.2 sec. per design iteration. Thus, the program based on the substructuring formulation is more efficient as compared to a program without substructuring formulation.

4.3.2 Closed Helicopter Tail-Boom

In this example, the same helicopter tail-boom structure as discussed in Section 4.3.1, is considered. The tail-boom is modeled as a closed structure that is obtained by using a skin cover on the 108 member truss of Figure 2.1. Design data for the structure is the same as given in Appendix A, except for the skin material. The skin is an aluminum alloy sheet (7075-T6 clad aluminum) that is modeled by 48 CST elements. The element connectivity for the CST elements is defined in Table 4.5. Material properties for the skin are: Young's Modulus = 10,400 ksi, yield stress = 67 ksi, working stress = 40.2 ksi, and the material weight density = 0.098 lbs/in^3.

TABLE 4.5. CST ELEMENT CONNECTIVITY FOR CLOSED HELICOPTER TAIL-BOOM

Member Number	Node i	Node j	Node k	
1	7	9	11	Substructure #1
2	7	9	5	
3	5	3	7	
4	5	3	1	
5	6	12	10	
6	6	12	8	
7	2	8	6	
8	2	8	4	
9	6	9	10	
10	6	9	5	
11	2	5	6	
12	2	5	1	
13	12	7	11	
14	12	7	8	
15	8	3	7	
16	8	3	4	
17	5	11	7	Substructure #2
18	5	11	9	
19	9	3	11	
20	9	3	1	
21	10	8	6	
22	10	8	12	
23	12	2	4	
24	12	2	10	
25	10	5	6	
26	10	5	9	
27	2	9	10	
28	2	9	1	
29	8	11	7	
30	8	11	12	
31	12	3	11	
32	12	3	4	
33	5	11	7	Substructure #3
34	5	11	5	
35	9	3	11	
36	9	3	1	
37	10	8	6	
38	10	8	12	
39	12	2	2	
40	12	2	10	
41	10	5	6	
42	10	5	9	
43	2	9	10	
44	2	9	1	
45	8	11	7	
46	8	11	12	
47	12	3	11	
48	12	3	4	

The structure is divided into three substructures for the computer program SOS4. Data for the three substructures are the same as discussed in Section 4.3.1, except that skin elements must be included in all substructures. The starting design for the structure is taken as 0.10 in^2. for the truss elements and 0.04 in. for the CST elements. Lower and upper bounds for the CST elements are 0.02 in. and 0.05 in., respectively. The lower bound for truss elements is the same as in the previous example.

Optimum solutions for Cases I and V of Appendix A are again obtained using the program SOS4. Results are given in Table 4.6. Considerable design variable linking is used in this example, as indicated in Table 4.6. Critical constraints at the optimum for this example problem are given in Table 4.7. Fundamental frequencies of the closed and open tail-boom structures are given in Table 4.8.

Comparison of Optimum Results for Open and Closed Tail-Boom Structures: Table 4.9 presents a comparison of optimum results for the open and closed tail-boom structures. For both Cases I and V, the optimum weight for the closed tail-boom is less than half that for the open tail-boom. Thus, one can conclude that the closed tail-boom is considerably more efficient in carrying loads.

Computational effort for the closed tail-boom is greater than that for the open tail-boom, since the closed tail-boom has more members and design variables than the open tail-boom.

TABLE 4.6. FINAL DESIGN FOR CLOSED HELICOPTER TAIL-BOOM WITH SUBSTRUCTURING

Design Variable Number	Member Number	CASE I (without damage) At 10th Iter.	CASE I (without damage) At 30th Iter.	CASE V (with 6 damage conditions)
1	2,3	0.0554	0.0415	0.0847
2	1,4	0.0542	0.0415	0.1498
3	5,6,9,10	0.0415	0.0415	0.0415
4	7,8,11,12	0.0438	0.0415	0.1138
5	13,15	0.0415	0.0415	0.0415
6	14,16	0.0415	0.0415	0.0415
7	17,18	0.0645	0.0415	0.0418
8	20,21	0.0631	0.0415	0.1885
9	19,22	0.0415	0.0415	0.3267
10	23,24,27,28	0.0415	0.0415	0.2522
11	25,26,29,30	0.0415	0.0415	0.0526
12	31,33	0.0415	0.0415	0.0415
13	32,34	0.0415	0.0415	0.0415
14	35,36	0.0415	0.0415	0.2405
15	38,39	0.0415	0.0415	0.0415
16	37,40	0.0415	0.0415	0.0415
17	41,42,45,46	0.0415	0.0415	0.0415
18	43,44,47,48	0.0486	0.0415	0.0415
19	49,51	0.0415	0.0415	0.0415
20	50,52	0.0415	0.0415	0.0415
21	53,54	0.0415	0.0415	0.0415
22	56,57	0.0415	0.0415	0.0415
23	55,58	0.0415	0.0415	0.0482
24	59,60,63,64	0.0415	0.0415	0.0415
25	61,62,65,66	0.0515	0.0415	0.0415
26	67,69	0.0415	0.0415	0.0415

TABLE 4.6 Cont'd

Design Variable Number	Member Number	CASE I (without damage)		CASE V (with 6 damage conditions)
		At 10^{th} Iter.	At 30^{th} Iter.	
27	68,70	0.0415	0.0415	0.0415
28	71,72	0.0415	0.0415	0.0624
29	74,75	0.0415	0.0415	0.0415
30	73,76	0.0415	0.0415	0.0415
31	77,78,81,82	0.0419	0.0415	0.0415
32	79,80,83,84	0.0553	0.0415	0.0429
33	85,87	0.0415	0.0415	0.0415
34	86,88	0.0415	0.0415	0.0415
35	89,90	0.0816	0.0814	0.1941
36	92,93	0.0415	0.0415	0.0437
37	91,94	0.0415	0.0415	0.0640
38	95,96,99,100	0.0415	0.0415	0.1822
39	97,98,101,102	0.0534	0.0415	0.1229
40	103,105	0.0415	0.0415	0.0415
41	104,106	0.0415	0.0415	0.0415
42	107,108	0.0415	0.0415	0.0820
43 (CST)	1-4	0.0385	0.04555	0.0374
44 (CST)	5-8	0.0313	0.02325	0.05
45 (CST)	9-16	0.02	0.02	0.0425
46 (CST)	17-20	0.0414	0.04709	0.05
47 (CST)	21-24	0.0285	0.02	0.05
48 (CST)	25-32	0.02	0.02	0.0420
49 (CST)	33-36	0.0434	0.04550	0.05
50 (CST)	37-40	0.02	0.02	0.0399
51 (CST)	41-48	0.02	0.02	0.0373
Weight in lbs.		45.82	44.53	77.81
Average CPU/iter. in sec. on IBM 370-158		7.66	7.66	52.00
# of active constrs. at optimum		34	48	35
$\|\|\delta b^1\|\|_{opt.}$		158.8	77.63	253.8
$\|\|\delta b^1\|\|_{max}$		321.9	321.9	434.3

TABLE 4.7. CRITICAL CONSTRAINTS AT OPTIMUM (CLOSED TAIL-BOOM)

CASE I: Without damage

(i) Results obtained after 10 design iterations (1 stress-ratio design cycle initially)

Lower bound on design variable numbers:
3, 5, 6, 7, 10-17, 19-21, 23, 24, 26-30, 33, 34, 36-38, 40-42, and 50, 51

(ii) Results obtained after 30 design iterations (1 stress-ratio design cycle initially)

- Displacement of nodes 1 and 3 in the x_2 direction of the 3rd substructure
- Lower bound on design variable numbers: 1-34, 36-42, 50, 51
- Max. violation is 0.31%

CASE V: With 6 damage conditions (results are obtained after 38 design iterations with no stress-ratio design initially)

- Displacement of nodes 1 and 3 in the x_2 direction of the 3rd substructure under damage conditions 2, 4, and 6
- Truss member #98, group 39 in substructure 3 under damage condition 5
- Lower bound on design variable numbers: 3, 5, 6, 12, 13, 15-22, 24-27, 40, 41, 29-31, 33-34
- Upper bound on design variable numbers: 44, 46, 47 and 49
- Max. violation is 1.04% (in displacement constraints)

TABLE 4.8. NATURAL FREQUENCY AT OPTIMUM (RESONANT FREQUENCY = 29 HZ.)

		Natural Frequencies (in Hz.)
	CASE I: Without damage	33.51
(Open tail-boom)	CASE V: With 6 damage conditions	43.91, 30.69, 28.999, 44,39, 46,93, 44.07, 28.999
	CASE I: Without damage	53.28
(Closed tail-boom)	CASE V: With 6 damage conditions	56.01, 36.68, 36.44, 60.64, 62.39, 60.84, 34.55

TABLE 4.9. COMPARISON OF OPTIMUM RESULTS OBTAINED WITH SUBSTRUCTURING FOR OPEN AND CLOSED TAIL-BOOM STRUCTURES

	CASE I		CASE V	
	Optimum Weight, lbs.	CPU Time per Design Iter.	Optimum Weight, lbs.	CPU Time per Design Iter.
Open Tail-Boom	106.0	5.65	161.7	43.5
Closed Tail-Boom	44.5	7.66	77.8	52.0

CHAPTER 5

DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

A general method for optimal design of fail-safe structures with substructuring is presented. The method integrates the state space design sensitivity analysis for fail-safe structural design into the gradient projection method of design optimization. A step-by-step algorithm for optimal design of fail-safe structures is stated. Computational aspects of the method with substructuring are discussed.

A modified Subspace Iteration method for computing natural frequencies of a structure by substructuring is also developed and integrated into the optimal design algorithm. The method is quite readily programmed and integrated into the optimization algorithm with substructuring. An analysis of the method indicates that there is a trade-off between computational time and computer core requirements, depending on whether the mass matrix for the structure is computed only once for Subspace Iteration or computed in every Subspace Iteration.

Comparing results for the open and closed tail-boom structures, one concludes that the closed structure is considerably lighter in weight to support a given set of loads. Thus, if two tail-booms are constructed - one open and the other closed- that weight roughly the same, the closed tail-boom can be designed so that it is able to withstand more damage than the open tail-boom. However, there is a trade-off between the open and the closed tail-boom structures. The trade-off is in vulnerability of the two structures to blast. Whereas the closed tail-boom is more efficient in load carrying and sustaining damage, it is more vulnerable to damage by projectile fragments and charge detonations inside the tail-boom structure. Also, the closed tail-boom has more exposed surface area that is vulnerable to damage. In comparison, the open tail-boom is less succeptable to damage because it has smaller exposed surface area. Also projectile or blast fragments may simply pass through the structure with any damage. These trade-offs should be more thouroughly analyzed before a decision is made to go ahead with either an open or a closed tail-boom.

There are several areas of research that need to be investigated to fully utilize potential of the optimal design method developed for fail-safe structures. These areas are briefly outlined here:

(1) Potential use of fiber-reinforced composite materials in fail-safe design of aircraft, helicopter, and other structures should be evaluated . Trade-offs between structural weight, damage sustainance, ease of construction, and construction costs should be analyzed.

(2) Definition of damage to a structure needs to be refined. In the present work, damage to parts of the structure is specified for the designer before he sizes the structural elements. However, prediction of a damage condition should take member sizes into consideration.

(3) The finite elements modeling of the structure needs to be refined. The finite element library for the computer program should be expanded to include elements such as the beam, the plate, and the quadrilateral membrane element. The disign sensitivity analysis method with these elements should be developed.

(4) The effect of body forces should be incorporated into the computer program. Also the effect of temperature variations on structural performance should be included in the computer program. Note that the general optimal design formulation and the algorithm already account for these effects.

(5) An algorithm for optimal design of fail-safe structures that use commercially available sections should be developed. This will reduce fabrication costs.

(6) A formulation and an optimization algorithm for fail-safe design of structure, subjected to transient dynamic loads should be developed.

(7) Work should continue in development of innovations for improving efficiency of the basic optimal design algorithm for fail-safe structures. Improvements in treatment of fail-safe constraints, step size selection techniques, and selection of weighting parameters are some of the areas that need further refinement.

LIST OF REFERENCES

1. Sun, P.F., Arora, J.S. and Govil, A.K., User's Manual for a Computer Program for Fail-Safe Design of Trusses, Technical Report No. 26, Materials Division, College of Engineering, The University of Iowa, Iowa City, Ia., Feb. 1977.

2. Haug, E.J. and Arora, J.S., Engineering Design Handbook: Computer Aided Design of Mechanical Systems Volume II, AMCP 706-193, U.S. Army Materiel Command, Washington, D.C., July 1977.

3. Govil, A.K., Arora, J.S. and Haug, E.J., Substructuring Methods for Design Sensitivity Analysis and Structural Optimization, Technical Report No. 34, Materials Division, College of Engineering, The University of Iowa, Iowa City, Ia., August 1977.

4. Przemieniecki, J.S., Theory of Matrix Structural Analysis, McGraw-Hill Book Co., Inc., New York, 1968.

5. Bathe, K.J., and Wilson, E.L., "Solution Methods for Eigenvalue Problems in Structural Mechanics," International Journal for Numerical Methods in Engineering, Vol. 6, No. 2, 1973, pp. 213-226.

6. Householder, A.S., Principles of Numerical Analysis, McGraw-Hill Book Co., New York, 1953.

7. Gupta, K.K., "Free Vibrations of Single-Branch Structural Systems," Journal of the Institute of Mathematics and Its Applications, Vol. 5, No. 3, 1969, pp. 351-362.

8. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Claredon Press, Oxford, 1965.

9. Peters, G., and Wilkinson, J.H., "Eigenvalues of $Ax = \lambda Bx$ with Band Symmetric A and B." Computer Journal, Vol. 12, 1969, pp. 398-404.

10. Ostrowski, A.M., Solution of Equations and Systems of Equations, Academic Press, New York, 1960.

11. Arora, J.S., "Survey of Structural Reanalysis Techniques," Journal of the Structural Division, Proc. ASCE, Vol. 102, No. ST4, April 1976, pp. 783-802.

12. Fox, R.L., and Kapoor, M.P., "A Minimization Method for the Solution of the Eigenproblem Arising in Structural Dynamics," presented at the Air Force Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, October, 15-17, 1968.

13. Bradbury, W.W., and Fletcher, R., "New Iterative Methods for Solution of the Eigenproblem," Numerishce Mathematik 9, 1966, pp. 259-267.

14. Arora, J.S., and Haug, E.J., Jr., "Efficient Optimal Design of Structures by Generalized Steepest Descent Programming," International Journal for Numerical Methods in Engineering, Vol. 10, No. 4, 1976, pp. 747-766, and Vol. 10, No. 6, 1976, pp. 1420-1426.

15. Haug, E.J. and Arora, J.S., "Design Sensitivity Analysis of Elastic Mechanical Systems," Computer Methods in Applied Mechanics and Engineering, to appear, 1978.

16. Arora, J.S. and Haug, E.J., Methods of Design Sensitivity Analysis in Structural Optimization, Technical Report No. 40, Materials Division, College of Engineering, The University of Iowa, Iowa City, Ia., May 1978.

17. Haug, E.J., Jr., Pan, K.C. and Streeter, T.D., "A Computational Method for Optimal Structural Design: I Piecewise Uniform Structures," International Journal for Numerical Methods in Engineering, Vol. 5, 1972, pp. 171-184.

18. Arora, J.S. and Govil, A.K., "An Efficient Method for Optimal Structural Design by Substructuring," International Journal of Computers and Structures, Vol. 7, No. 4, 1977, pp. 507-515.

19. Arora, J.S. and Govil, A.K., "Design Sensitivity Analysis with Substructuring," Journal of Engineering Mechanics Division, Proceedings of the ASCE, Vol. 103, No. EM4, August 1977, pp. 537-548.

20. Ugural, A.C. and Fenster, S.K., Advanced Strength and Applied Elasticity, American Elsevier Publishing Co., Inc., New York, 1975.

APPENDIX A

to

Report Number 45

FAIL-SAFE DESIGN OF AN OPEN TRUSS HELICOPTER TAIL-BOOM WITHOUT SUBSTRUCTURING

The purpose of this appendix is to present optimal designs for several cases of an open truss helicopter tail-boom without substructuring. The structure is the tail-boom for a U. S. Army Cobra helicopter.

The basic configuration and end sections of the tail-boom are shown in Figure A.1. The maximum in-flight loads to be supported by the tail-boom structure are also shown in Figure A.1. The structure that is currently in use consists of longitudinal members, cross members, and a skin cover to obtain an enclosed tail-boom. This type of structure is vulnerable to blasts that occur inside or near the skin. In order to reduce vulnerability of the structure to such damage, an open truss type structure is considered. Accordingly, the structure shown in Figure A.1 is modeled as a 108 member truss with 28 joints and 72 degrees of freedom. The geometry of the idealized structure and the design loads are given in Figure A.2. The element numbering system for a typical panel is shown in Figure A.3. The member definitions for the truss are given in Table A.1.

The problem is to minimize the total weight of the structure and at the same time to ensure that member stress, nodal displacement, member buckling, and natural frequency constraints are satisfied under projected loading and damage conditions. The design parameters to be calculated are the cross-sectional areas of the members. A lower bound constraint is also imposed on cross-sectional area.

The members of the truss are taken to be tubular sections. Assuming the sections to be thin, the moment of inertia and cross-sectional area are given as $I = \pi R^3 t$ and $A = 2\pi Rt$, where R is the mean radius and t is the thickness of the tube. In calculating the Euler buckling load, the moment of inertia is assumed to be given as $I = \bar{\alpha} A^2$. Therefore, $\bar{\alpha} = I/A^2$ is given as $R/4\pi t$. If R/t is conservatively selected as 12 to 14, then $\bar{\alpha} \simeq 1.0$. This value of $\bar{\alpha}$ is used in calculations.

Design data for the structure are given in Table A.2. The working stress for each member is assumed to be approximately 60 percent ($\pm$ 25 ksi) of the yield stress (42 ksi) for the material used. This working stress corresponds to a safety factor of roughly 1.68. Displacement limit of $\pm$ 0.5 in. at the nodal points are based on approximately $1/3^{\circ}$ mis-alighment at the center of the tail-boom. The lower limit on member cross-sectional area is taken as 0.0415

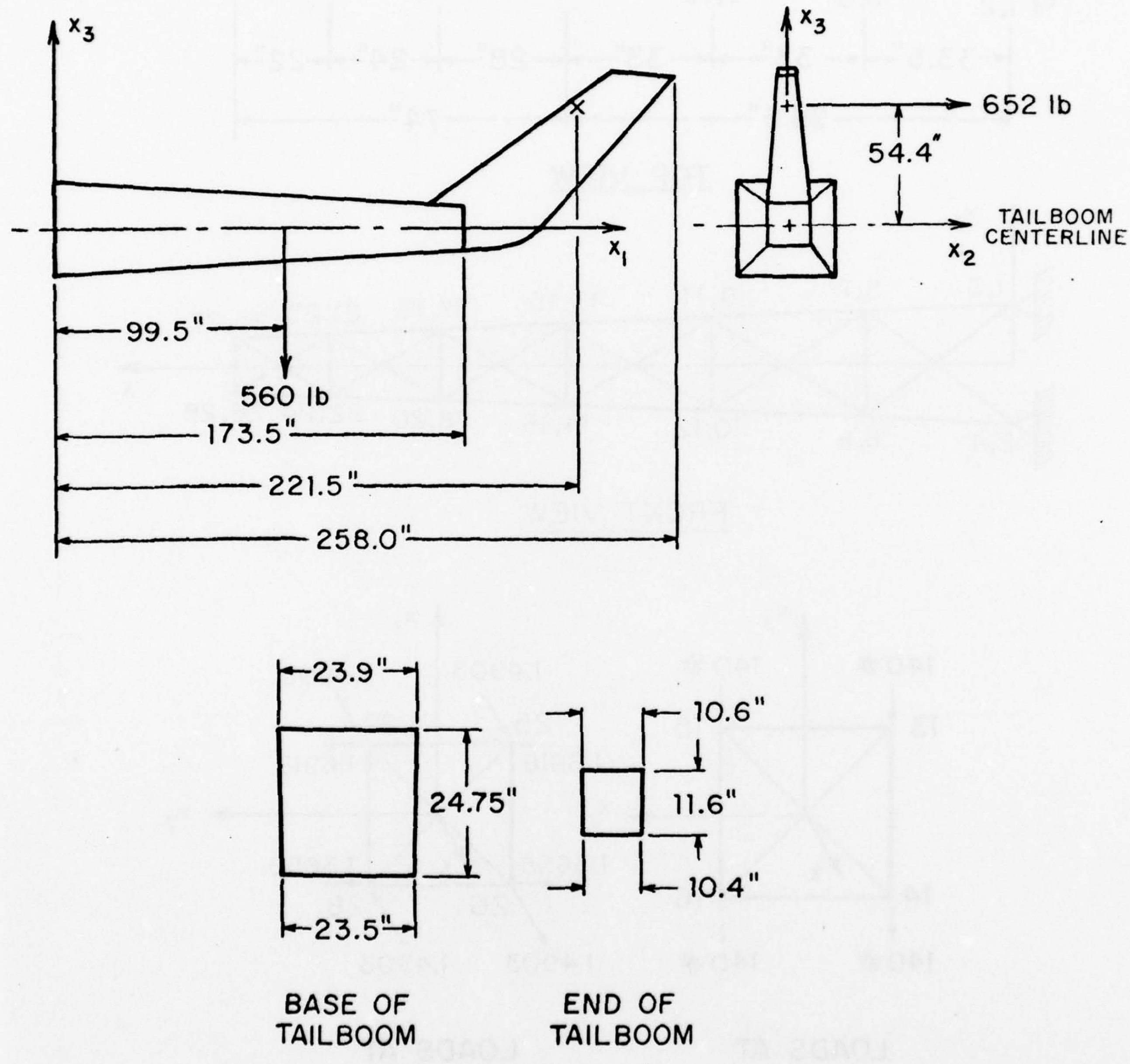


Figure A.1. Geometry of Helicopter Tail Boom

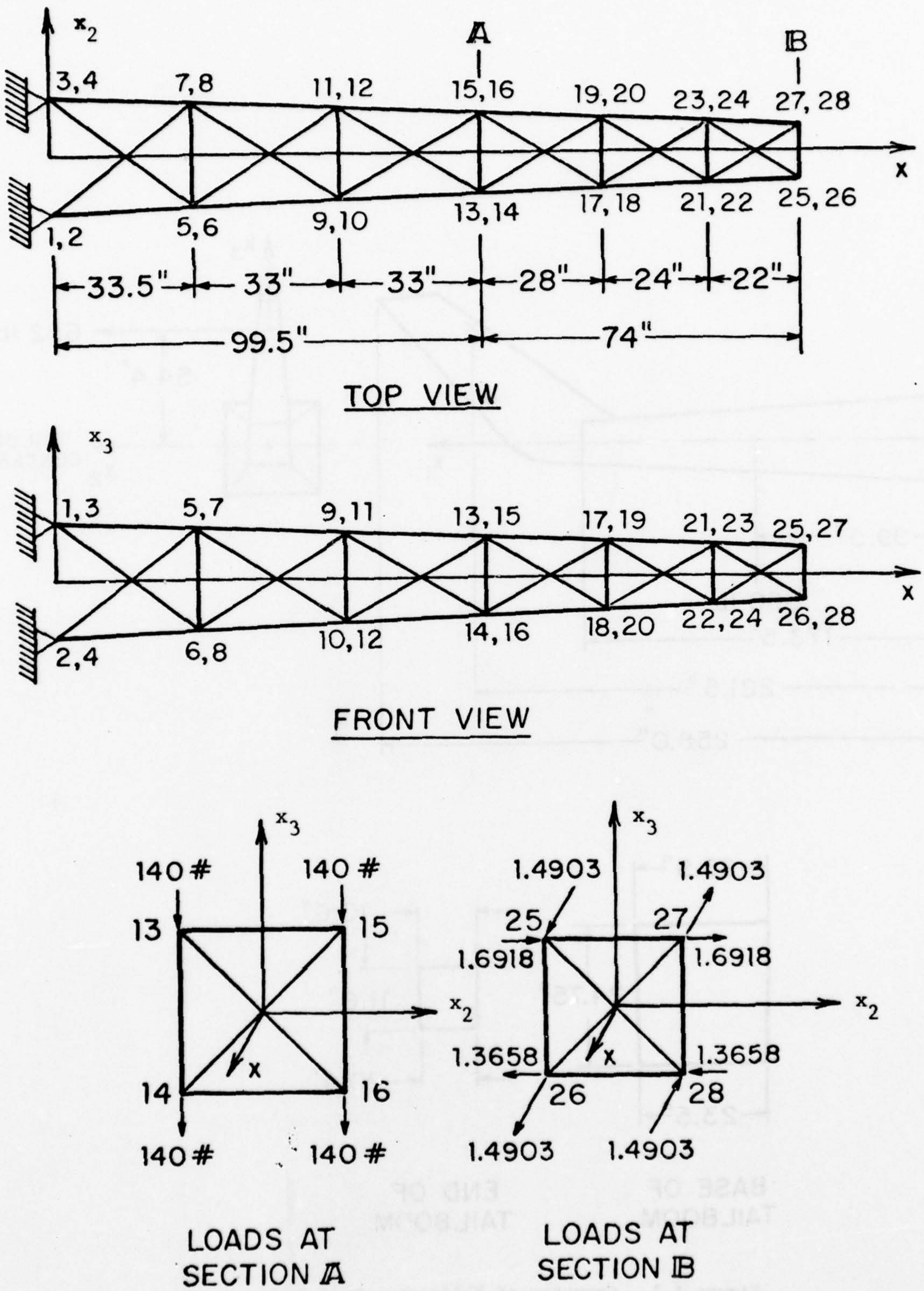
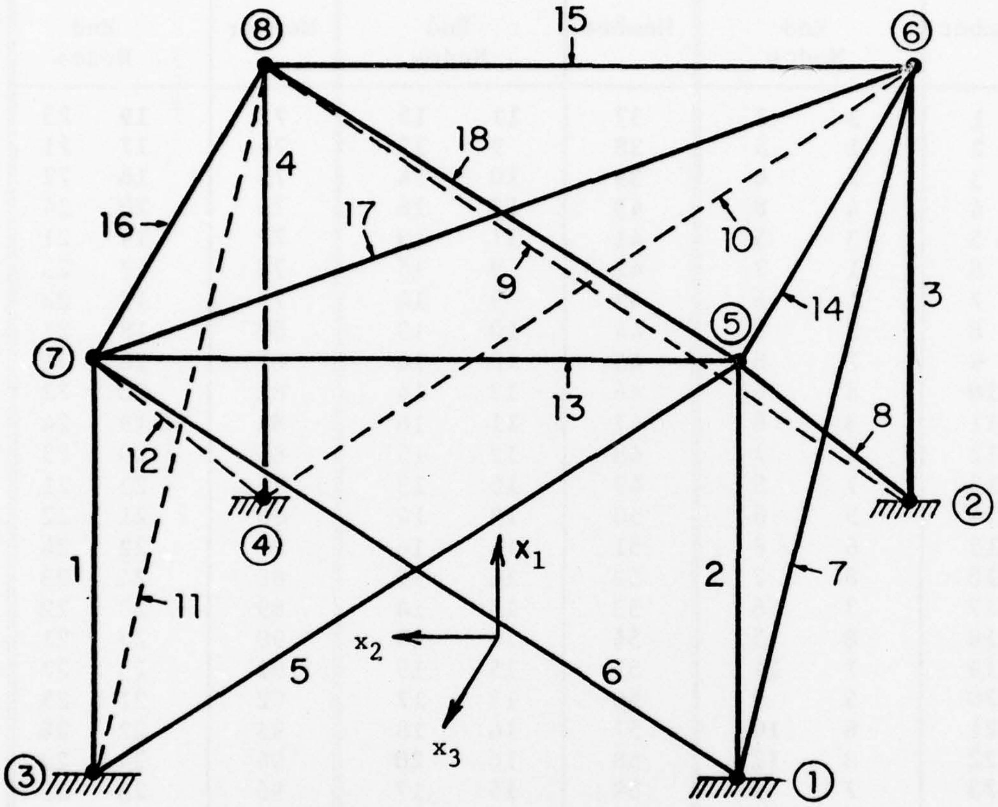


Figure A.2. Arrangement of Members for Open Truss Tail-Boom



Following grouping of members with members of a group required to have same cross-sectional areas is used to maintain some symmetry in the structure

Group No.	Member Numbers
1	2,3
2	1,4
3	5,6,9,10
4	7,8,11,12
5	13,15
6	14,16
7	17,18

Figure A.3. Member Numbering for the First Panel

TABLE A.1

MEMBER LOCATIONS FOR OPEN TRUSS HELICOPTER TAIL BOOM

Member	End Nodes		Member	End Nodes		Member	End Nodes	
1	3	7	37	11	15	73	19	23
2	1	5	38	9	13	74	17	21
3	2	6	39	10	14	75	18	22
4	4	8	40	12	16	76	20	24
5	3	5	41	11	13	77	19	21
6	1	7	42	9	15	78	17	23
7	1	6	43	9	14	79	17	22
8	2	5	44	10	13	80	18	21
9	2	8	45	10	16	81	18	24
10	4	6	46	12	14	82	20	22
11	3	8	47	11	16	83	19	24
12	4	7	48	12	15	84	20	23
13	7	5	49	15	13	85	23	21
14	5	6	50	13	14	86	21	22
15	6	8	51	14	16	87	22	24
16	8	7	52	16	15	88	24	23
17	7	6	53	15	14	89	23	22
18	8	5	54	16	13	90	24	21
19	7	11	55	15	19	91	23	27
20	5	9	56	13	17	92	21	25
21	6	10	57	14	18	93	22	26
22	8	12	58	16	20	94	24	28
23	7	9	59	15	17	95	23	25
24	5	11	60	13	19	96	21	27
25	5	10	61	13	18	97	21	26
26	6	9	62	14	17	98	22	25
27	6	12	63	14	20	99	22	28
28	8	10	64	16	18	100	24	26
29	7	12	65	15	20	101	23	28
30	8	11	66	16	19	102	24	27
31	11	9	67	19	17	103	27	25
32	9	10	68	17	18	104	25	26
33	10	12	69	18	20	105	26	28
34	12	11	70	20	19	106	28	27
35	11	10	71	19	18	107	27	26
36	12	9	72	20	17	108	28	25

TABLE A.2

DESIGN DATA FOR OPEN TRUSS HELICOPTER TAIL BOOM

A: Data Common to complete as well as damaged structures

Material	:	2024-T3 Aluminum Alloy
Modulus of Elasticity	=	10.5×10^3 ksi
Stress limits	=	± 25.0 ksi
Material density	=	0.1 lb/in.^{3}
Moment of inertia	:	$I = \overline{\alpha}A^2$; $\overline{\alpha} = 1.0$
Displacement limits	=	± 0.50 in.
Lower limit on cross-Sectional area	=	0.0415 in.^{2}
Upper limit on cross-Sectional area	=	None

B: Loading Data

Number of loading conditions = one

Loading for complete structure :

Node Number	Load Component (kips) in direction		
	X_1	X_2	X_3
13	0.0	0.0	-0.140
14	0.0	0.0	-0.140
15	0.0	0.0	-0.140
16	0.0	0.0	-0.140
25	1.4903	1.6918	0.0
26	1.4903	-1.3658	0.0
27	-1.4903	1.6918	0.0
28	-1.4903	-1.3658	0.0

Lower bound on natural frequency for complete structure = 29 Hz

in.^{2} which corresponds to a tube with 0.50 in. outside diameter and 0.028 in. wall thickness. There is no upper limit on cross-sectional area. There is only one loading condition for the structure, which is given in Table A.2. There are six projected damage conditions for the structure, given in Table A.3. For each damage condition a joint of the structure and all members connected to the joint are removed. Thus each damaged structure has different stiffness and mass matrices and state variables. Note, however, that each damaged structure is geometrically stable.

In order to maintain symmetry and to facilitate fabrication of the structure, 108 members of the structure are divided into a total of 42 groups and each group is assigned a design variable. Therefore each panel of the structure (shown in Figure A.3) has seven design variables. Also, it is interesting to study the effect on structural weight obtained by imposing varying degrees of performance requirements for the damaged structures. Thus optimum solutions for the following five cases are obtained.

<u>Case I:</u> Complete structure with no damage.

<u>Case II:</u> Complete structure with damage conditions 1 to 6 imposed and the structural load and natural frequency requirements for damaged structures reduced to two-thirds of the normal conditions.

<u>Case III:</u> Same as Case II except load and natural frequency requirements for damaged structures are 80% of the normal conditions.

<u>Case IV:</u> Same as Case II except load and natural frequency requirements for damaged structures are 90% of the normal conditions.

<u>Case V:</u> Complete and damaged structures required to perform for full set of normal conditions.

Optimum designs for the open truss helicopter tail-boom for Cases I to V are given in Table A.4. These designs were obtained by starting the iterative process with 1.0 m^2 as cross-sectional area for all members of the tail-boom. Comparing the results for Cases I and II, one concludes that when performance requirements for projected damaged structures (defined in Table A.3) are reduced to two-thirds of the normal conditions there is essentially no penalty on the weight of the structure. However, there is some redistribution

TABLE A.3. DAMAGE CONDITION DEFINITIONS AND FREQUENCY LIMITS

Damage Condition	Member(s) Damaged	Node(s) Damaged	% Reduction in Area
1	21,25,28,32,33, 35,39,44,45	10	100
2	1,6,12,13,16, 17,19,23,29	7	100
3	58,63,65,69,70, 72,76,82,84	20	100
4	73,78,84,85,88, 89,91,95,101	23	100
5	56,59,62,67,68, 72,74,78,79	17	100
6	3,7,10,14,15, 17,21,26,27	6	100

of the material, as may be seen from optimal solutions for Cases I and II given in Table A.4. If the final design for Case I given in Table A.4 is taken as the starting design for Case II, there are large constraint violations. This indicates that the structure constucted from the solution of Case I would fail catastrophically if any of the damage conditions defined in Table A.3 occured, even after the load and the natural frequency requirements were reduced to two-thirds of the normal conditions. On the other hand, if a tail-boom is constructed from the final areas for Case II, the structure is able to safely support two-thirds of the load carrying requirement, even after any of the specified damage occurs.

Final designs for Cases III, IV, and V are also given in Table A.4. They indicate that there is a substantial penalty on the weight of the structure as the load carrying and natural frequency requirements for the damaged structures are increased.

Due to ease in fabrication, it is desirable to use as few standard sections as possible. For the design of Cases I to V, 42 design variables (that is 42 types of sections) are used. This number is perhaps too large. Therefore tail-boom design for two additional cases VI and VII is also obtained. These cases are as follows:

<u>Case VI:</u> The number of design variables is reduced to 12, with 2 design variables for each bay. For the first bay of Figure A.3, members 1-4, 13-16 have same cross-sectional areas and members 5-12, 17 and 18 have same cross-sectional areas. The tail-boom is designed with six damage conditions of Table A.3 imposed, and complete and damaged structures are required to perform for full set of normal conditions.

<u>Case VII:</u> The number of design variables is reduced to 4 with 2 design variables for first three bays and 2 design variables for the last three bays. For the first three bays, all longerons, vertical and cross members have the same cross-sectional areas and all diagonals have same cross-sectional areas. A similar grouping is done for the last three bays. The tail-boom is designed with six damage conditions of Table A.3 imposed, and complete and damaged structures are required to perform for full set of normal conditions.

TABLE A.4.

OPTIMUM DESIGNS FOR CASES I TO V OF THE TAIL-BOOM STRUCTURE

Design Variable	Member Numbers	Final Cross-Sectional Areas (in.^{2})				
		Case I	Case II	Case III	Case IV	Case V
1	2,3	1.3750	1.415	1.6930	2.2810	3.0250
2	1,4	1.3710	1.424	1.6440	2.1220	2.7880
3	5,6,9,10	0.1375	0.1391	0.2094	0.2427	0.2684
4	7,8,11,12	0.1395	0.1544	0.1464	0.1589	0.2266
5	13,15	0.0415	0.0415	0.0415	0.0726	0.0998
6	14,16	0.0821	0.0809	0.1374	0.1700	0.1589
7	17,18	0.0415	0.0415	0.1777	0.3187	0.3168
8	20,21	1.2420	1.2610	1.3870	1.7060	2.2440
9	19,22	1.2390	1.2600	1.2400	1.5770	2.0930
10	23,24,27,28	0.1741	0.1593	0.1751	0.2161	0.3964
11	25,26,29,30	0.1649	0.1864	0.3504	0.3981	0.4220
12	31,33	0.0415	0.0415	0.0479	0.0415	0.0477
13	32,34	0.1002	0.1034	0.1948	0.1972	0.1522
14	35,36	0.0415	0.0498	0.0909	0.1035	0.1231
15	38,39	1.0290	1.022	1.0550	1.1060	1.3060
16	37,40	1.0280	1.0070	1.0040	1.070	1.2700
17	41,42,45,46	0.2110	0.1990	0.2301	0.2585	0.3404
18	43,44,47,48	0.2295	0.2513	0.2464	0.2738	0.2894
19	49,51	0.0415	0.0415	0.0498	0.0818	0.0928
20	50,52	0.1371	0.1315	0.1228	0.0962	0.0899
21	53,54	0.0415	0.0415	0.0451	0.0773	0.0897
22	56,57	0.8221	0.8218	0.8237	0.8759	0.9313
23	55,58	0.8226	0.8020	0.8179	0.9044	0.9761
24	59,60,63,64	0.2365	0.2316	0.3045	0.3733	0.4043
25	61,62,65,66	0.2587	0.2425	0.1891	0.1508	0.1583
26	67,69	0.0415	0.0415	0.0415	0.0753	0.0902
27	68,70	0.1575	0.1372	0.1715	0.1230	0.1078
28	71,72	0.0415	0.0503	0.1283	0.1745	0.1714

The optimal designs for the last two cases are also obtained using the same computer code [1] and by starting from uniform cross-sectional areas of 1.0 in.^{2} for all members. The final areas for Case VI are given in Table A.5 and for the Case VII, they are given in Table A.6. As expected, there is a substantial penalty in weight of the structure, as compared to the weight obtained in Case V. This indicates that the designer has to decide whether the weight of the structure or its fabrication cost is critical, because as the number of design variables is reduced the optimum weight of the structure increases.

The constraints that are critical at the optimum for all cases are given in Table A.7. For all cases, all active constraints are satisfied to within 0.10% of their allowable values. The natural frequencies of the complete and damaged structures at the optimum solution are given in Table A.8. The cost function histories for all cases are given in Figure A.4. In most cases, an optimum design or a design very close to the optimum was obtained in 20-30 iterations.

The rate of convergence to the optimum is highly dependent on proper selection of the step size parameter η. In order to see how critical the step size parameter is, several step sizes for Case II of the helicopter tail-boom were tried and it was possible to obtain convergence to the optimum in 20 iterations, as compared to 32 iterations shown in Figure A.4. The step size in all calculations was selected based on the idea of specifying a desired reduction in the cost function [2] . Change in the cost function is given by the linearized formula

$$\delta\psi_0 = \Lambda^{J^T}\delta b \tag{A.1}$$

Now substituting for $\delta\psi_0 = -\bar{r}\psi_0$ (where $\bar{r}$ is a specified reduction ratio and ψ_0 is the current value of the cost function) and for $\delta b = -\eta\delta b^1$ from Equation 2.5-8 (where δb^2 is assumed to zero; that is all constraints are assumed to be satisfied) into Equation 2.4-26, one obtains

$$\eta = \bar{r}\psi_0/\Lambda^{J^T}\delta b^1 \tag{A.2}$$

This formala is used in calculating the step size at the start of the iterations. The step size parameter is monitored and sometimes adjusted as the iterations progress.

TABLE A.4. (cont.)

Design Variable	Member Numbers	Final Cross-Sectional Areas (in.^{2})				
		Case I	Case II	Case III	Case IV	Case V
29	74,75	0.5806	0.5846	0.4995	0.5390	0.5515
30	73,76	0.5830	0.5689	0.5626	0.6633	0.6666
31	77,78,81,82	0.2675	0.2626	0.2331	0.2651	0.2934
32	79,80,83,84	0.2883	0.2695	0.3453	0.3273	0.3111
33	85,87	0.0415	0.0415	0.0449	0.0416	0.0580
34	86,88	0.1934	0.1676	0.2132	0.1705	0.1573
35	89,90	0.0415	0.0415	0.0544	0.1069	0.1223
36	92,93	0.2299	0. 2244	0.2274	0.2682	0.2740
37	91,94	0.2090	0.2250	0.2021	0.1372	0.1184
38	95,96,99,100	0.3295	0.3188	0.2905	0.2134	0.1855
39	97,98,101,102	0.3428	0.3248	0.3318	0.3382	0.3327
40	103,105	0.0564	0.0415	0.0757	0.0921	0.1089
41	104,106	0.1036	0.0875	0.0999	0.0947	0.0926
42	107,108	0.1987	0.1929	0.1905	0.1899	0.1822
Weight in pounds		105.6	105.8	116.8	134.8	161.1
Average CPU/Iter. in sec. on IBM 370-158(G)		4.0	24.0	26.4	26.7	26.7
Number of Active Constraints at Opt.		12	14	11	14	10
$\|\|\delta b^1\|\|$ at opt.		2.8	0.70	3.78	3.72	3.49
$\|\|\delta b^1\|\|$ max.		53.8	53.8	53.8	53.8	53.8

TABLE A.5.

OPTIMAL DESIGN FOR CASE VI OF HELICOPTER TAIL-BOOM

Design Variable	Member Numbers	Final Areas (in^2)
1	1-4, 13-16	2.9370
2	5-12, 17, 18	0.5698
3	19-22, 31-34	2.0430
4	23-30, 35-36	0.8459
5	37-40, 49-52	1.0760
6	41-48, 53, 54	0.4047
7	55-58, 67-70	0.7033
8	59-66, 71, 72	0.3615
9	73-76, 85-88	0.4470
10	77-84, 89,90	0.3294
11	91-94, 103-106	0.1554
12	95-102, 107-108	0.2511
Optimum Weight in pounds		241.57
Average CPU/cycle in sec. on IBM 370-158 (G)		26.8
Number of Active Constraints at Opt.		5
$\|\|\delta b^1\|\|$ at Opt.		6.1
$\|\|\delta b^1\|\|$ max.		89.7

TABLE A.6.

OPTIMAL DESIGN FOR CASE VII OF HELICOPTER TAIL-BOOM

Design Variable	Member Numbers	Final Areas (in.^{2})
1	1-4, 13-16, 19-22, 31-34, 37-40, 49-52	3.2960
2	5-12, 17, 18, 23-30, 35, 36 41-48, 53, 54	0.8895
3	55-58, 67-70, 73-76, 85-88, 91-94, 103-106	0.4283
4	59-66, 71, 72, 77-84, 89, 90 95-102, 107, 108	0.2796
Optimum Weight in pounds		346.25
Average CPU/cycle in sec. on IBM 370-158 (G)		18.0
Number of Active Constraints at opt.		3
$\|\|\delta b^1\|\|$ at opt.		0.035
$\|\|\delta b^1\|\|$ max		155.1

TABLE A.7.

CRITICAL CONSTRAINTS AT OPTIMUM

Case I

Displacement in the x_2 direction at nodes 25 and 27, and lower limit on design variable numbers 5, 7, 12, 14, 19, 21, 26, 28, 33, and 35.

Case II

Same as in Case I, except design variables 14 and 28 are not at their lower bounds and 40 is at its lower bound, and buckling constraint for members 18, 36, 71 are tight under damage conditions 6, 1, and 5, respectively.

Case III

Displacement in the x_2 direction at node 25 under damage conditions 1, 2, 4, 5 and 6, displacement in the x_2 direction at node 27 under damage conditions 1, 2, 5 and 6, and lower bound on design variables 5 and 26.

Case IV

Displacement in the x_2 direction at node 25 under damage conditions 1, 2, 3, 4 and 5, displacement in the x_2 direction at node 27 under damage conditions, 1, 2, 3, and 5, frequency constraints under damage conditions 2 and 6, buckling constraint for member 66 under damage condition 3, and lower bound on design variables 12 and 33.

Case V

Displacement in the x_2 direction at node 25 under damage conditions 2, 3, 4 and 5, displacement in the x_2 direction at node 27 under damage conditions 2, 3 and 5; frequency constraints under damage conditions 2 and 6; buckling constraint for member 66 under damage condition 3.

Case VI

Displacement in the x_2 direction at nodes 25 and 27 under damage conditions 4 and 5, and frequency under damage condition 2.

Case VII

Displacement in the x_2 direction at nodes 25 and 27 under damage condition 5 and frequency constraint under damage condition 2.

TABLE A.8.

STRUCTURAL FREQUENCY AT OPTIMUM

Damaged Condition	Frequency at Optimum (Hz)						
	Case I	Case II	Case III	Case IV	Case V	Case VI	Case VII
0*	34.34	34.90	36.75	39.80	44.12	42.52	43.19
1	-	24.83	26.00	27.44	30.85	31.79	35.08
2	-	22.06	23.82	26.10	29.00	29.00	29.00
3	-	35.61	37.58	40.81	44.70	43.56	41.55
4	-	37.62	39.64	42.81	47.21	45.81	45.77
5	-	35.52	37.38	40.53	44.40	43.46	41.36
6	-	22.42	23.85	26.10	29.00	29.41	29.42

* Complete Structure

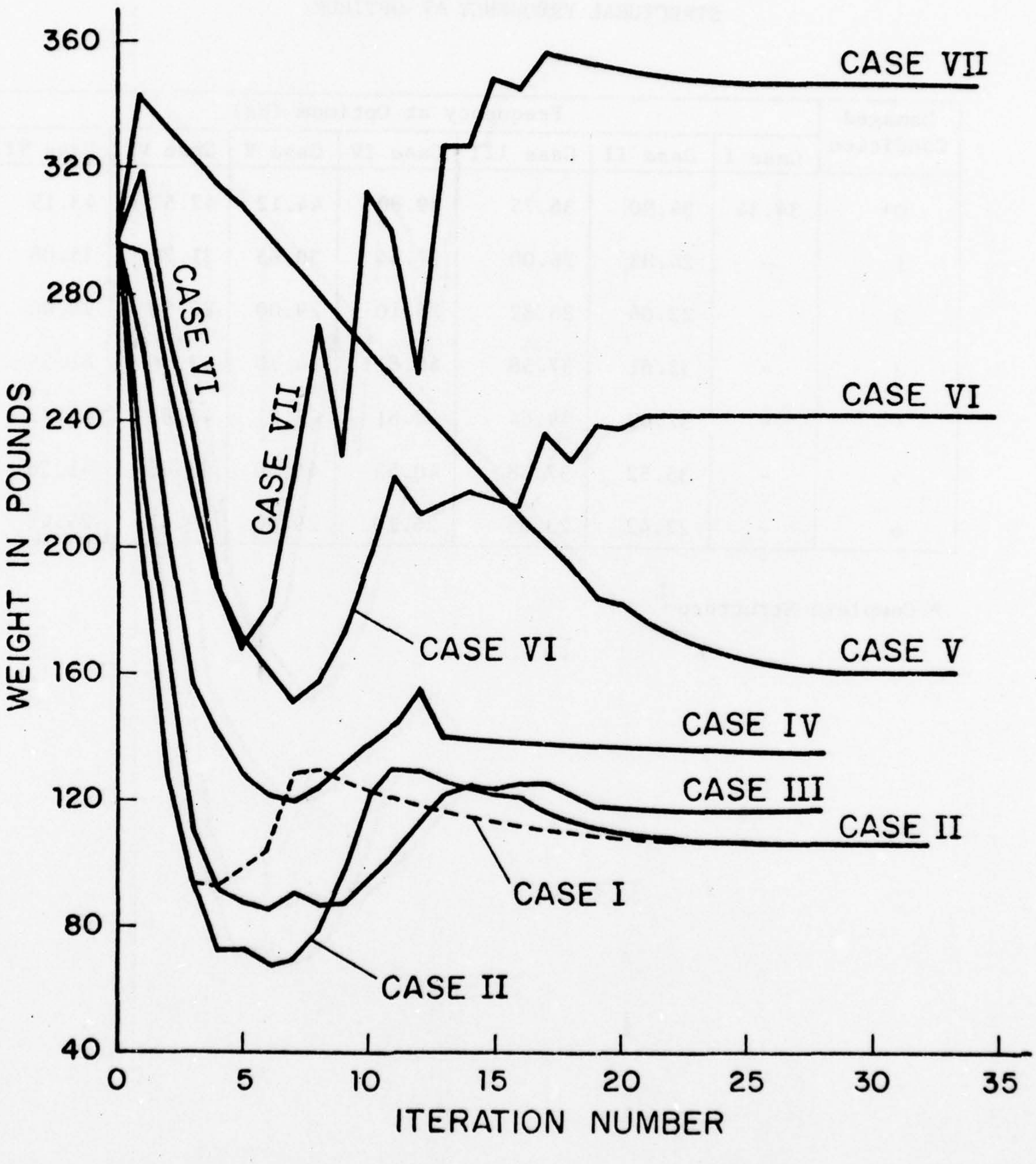


Figure A.4. Cost Function Histories for Several Design Cases of the Helicopter Tail-Boom Truss

It should be noted that in the first few design iterations for all cases of the tail-boom design, there were a large number of violations (50 to 100) and the maximum amount of violations was of the order of 1500%. The fail-safe optimal structural design algorithm corrected these constraints violations without difficulty.

APPENDIX B

to

Report Number 45

FINITE ELEMENTS EMPLOYED

The computer program for fail-safe structural optimazation with substructuring (FSOS) employs truss, constant strain triangular (CST), the symmetric shear panel (SSP) and symmetric pure shear panel (SPSP) finite elements. For convenience the stiffness and mass matrices for these elements are given in this appendix.

B.1. Notation and General Expressions

a = length of SSP or SPSP element

b = height of SSP or SPSP element

E = modulus of elasticity

ρ = material mass density

ℓ_1, m_1, n_1 = direction cosines of the local x_1 axis in the global coordinate system

$\tilde{u}, \tilde{v}, \tilde{w}$ = displacements in local coordinate system

x_1, x_2, x_3 = global coordinate system

$\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ = local coordinate system

$\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}$ = strain components in local coordinate system

$\sigma_{11}, \sigma_{22}, \sigma_{12}$ = stress components in local coordinate system

t = thickness of SSP or SPSP element

Θ = aspect ratio of SSP or SPSP element ($\Theta = \frac{a}{b}$)

$\tilde{r}$ = vector of nodal displacement in local coordinate system

B = strain-displacement relation matrix

$C, \tilde{C}$ = stress-displacement relation matrices in datum and local coordinate systems, respectively

D = stress-strain relation matrix

$k, \tilde{k}$ = element stiffness matrices in datum and local coordinate systems, respectively

R = rotation matrix from local to global coordinate system

β = local to global coordinate transformation matrix for stiffness and mass matrices

N = shape function that depends on $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$

V = volume of the finite element

A general expression for the element stiffness matrix in the local coordinate system is given as [4]:

$$\tilde{k} = \int_V B^T D B dV \qquad \text{(B.1-1)}$$

The element stiffness matrix relative to a global coordinate system can be obtained using β and $\tilde{k}$ matrices as follows [4].

$$k = \beta^T \tilde{k} \beta \qquad \text{(B.1-2)}$$

A general expression for the element mass matrix in the local coordinate system is given as [4]:

$$\tilde{m} = \int_V N^T N dV \qquad \text{(B.1-3)}$$

The mass matrix $\tilde{m}$ relative to a global coordinate system can be obtained according to the following prescription [4]:

$$m = \beta^T \tilde{m}\ \beta \qquad \text{(B.1-4)}$$

B.2. Truss Element

Truss is a one dimensional element that has constant strain throughout its length. Figure B.1 shows a general truss element in its local and global coordinate systems. Using the constant strain condition, shape functio for the truss element is given as [4]:

$$N = \begin{bmatrix} (1-\xi) & 0 & 0 & \xi & 0 & 0 \\ 0 & (1-\xi) & 0 & 0 & \xi & 0 \\ 0 & 0 & (1-\xi) & 0 & 0 & \xi \end{bmatrix} \qquad \text{(B.2-1)}$$

where $\xi = x_1/L$. Using Equation B.1-1, the stiffness matrix for the truss element is given as:

$$\tilde{k} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & \text{symmetric} & & 1 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad \text{(B.2-2)}$$

Using Equation B.1-3, mass matrix for the truss element is given as:

$$\tilde{m} = \frac{\rho AL}{6} \begin{bmatrix} 2I_3 & I_3 \\ I_3 & 2I_3 \end{bmatrix} \qquad \text{(B.2-3)}$$

where I_3 is a 3x3 identity matrix. It is shown in Ref. 4 that the mass matrix $\tilde{m}$ for the truss element is invariant under any rotation of the coordinate system, so $m = \tilde{m}$ for the truss element.

It can be easily shown that the stiffness matrix for the truss element relative to a global coordinate system can be expressed as:

$$k = [\frac{AE}{L}]\beta^T \beta \qquad \text{(B.2-4)}$$

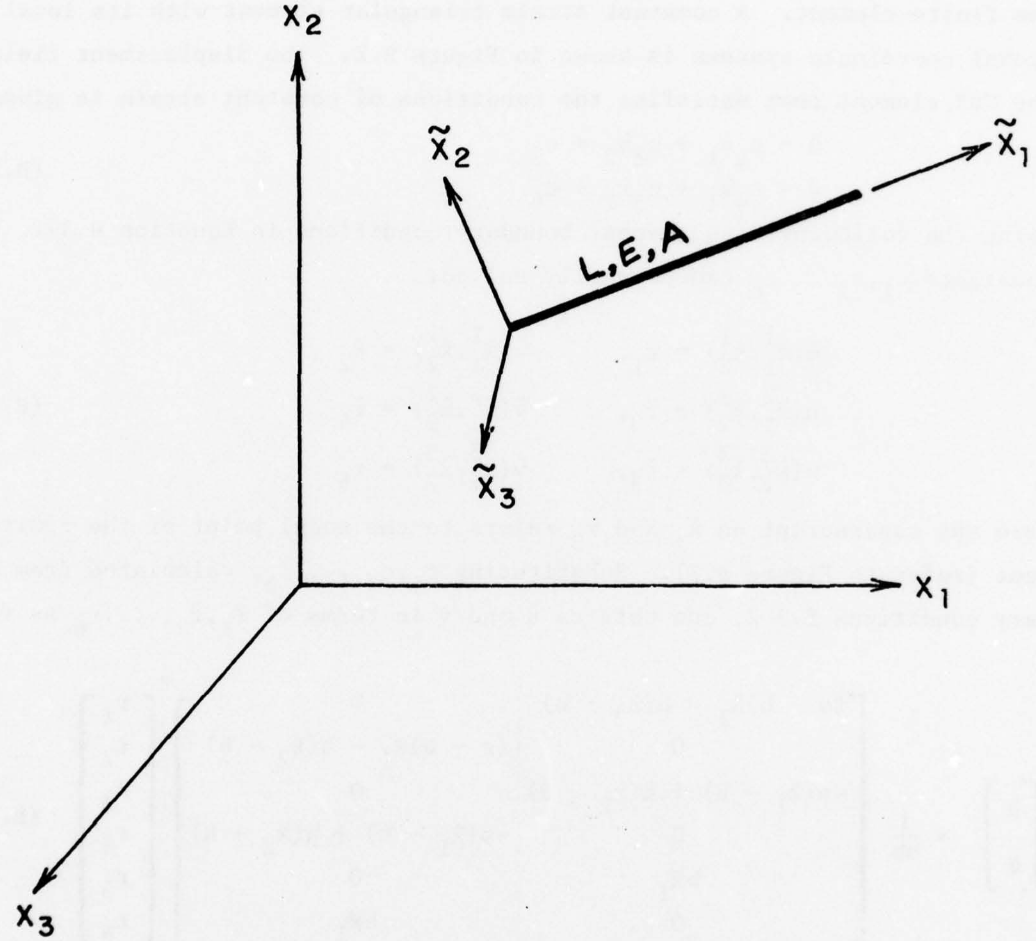


Figure B.1. A General Truss Element

where the row vector β is given as:

$$\beta = \begin{bmatrix} \ell_1 & m_1 & n_1 & -\ell_1 & -m_1 & -n_1 \end{bmatrix} \tag{B.2-5}$$

B.3. Isotropic Constant Strain Triangular (CST) Element

This element resists only in-plane stresses, σ_{11}, σ_{22}, and σ_{12}. These stresses and the corresponding strains are assumed to be constant throughout the finite element. A constant strain triangular element with its local and global coordinate systems is shown in Figure B.2. The displacement field for the CST element that satisfies the conditions of constant strain is given as:

$$\begin{aligned} \tilde{u} &= c_1\tilde{x}_1 + c_2\tilde{x}_2 + c_3 \\ \tilde{v} &= c_4\tilde{x}_1 + c_5\tilde{x}_2 + c_6 \end{aligned} \tag{B.3-1}$$

Using the following displacement boundary conditions in Equation B.3-1, the constants $c_1, c_2 \ldots . c_6$ can be easily solved:

$$\begin{aligned} \tilde{u}(\tilde{x}_1^1, \tilde{x}_2^1) &= \tilde{r}_1, & \tilde{v}(\tilde{x}_1^1, \tilde{x}_2^1) &= \tilde{r}_2 \\ \tilde{u}(\tilde{x}_1^2, \tilde{x}_2^2) &= \tilde{r}_3, & \tilde{v}(\tilde{x}_1^2, \tilde{x}_2^2) &= \tilde{r}_4 \\ \tilde{u}(\tilde{x}_1^3, \tilde{x}_3^3) &= \tilde{r}_5, & \tilde{v}(\tilde{x}_1^3, \tilde{x}_2^3) &= \tilde{r}_6 \end{aligned} \tag{B.3-2}$$

Here the superscript on $\tilde{x}_1$ and $\tilde{x}_2$ refers to the nodal point of the finite element (refer to Figure B.2). Substituting $c_1, c_2, \ldots . c_6$, calculated from boundary conditions B.3-2, one obtains $\tilde{u}$ and $\tilde{v}$ in terms of $\tilde{r}_1, \tilde{r}_2, \ldots . \tilde{r}_6$ as follows.

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \frac{1}{bh} \begin{bmatrix} (s-b)\tilde{x}_1 - h(\tilde{x}_2 - b) & 0 \\ 0 & (s-b)\tilde{x}_1 - h(\tilde{x}_2 - b) \\ -s(\tilde{x}_1 - h) + h(\tilde{x}_2 - s) & 0 \\ 0 & -s(\tilde{x}_1 - h) + h(\tilde{x}_2 - h) \\ b\tilde{x}_1 & 0 \\ 0 & b\tilde{x}_1 \end{bmatrix}^T \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{bmatrix} \tag{B.3-4}$$

From Equation B.3-3, the shape function N can be identified for the isotropic CST element.

The strains for the isotropic CST element are given as:

$$\varepsilon \equiv \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix} = \begin{bmatrix} u,_1 \\ v,_2 \\ u,_2 + v,_1 \end{bmatrix} \equiv B\tilde{r} \tag{B.3-4}$$

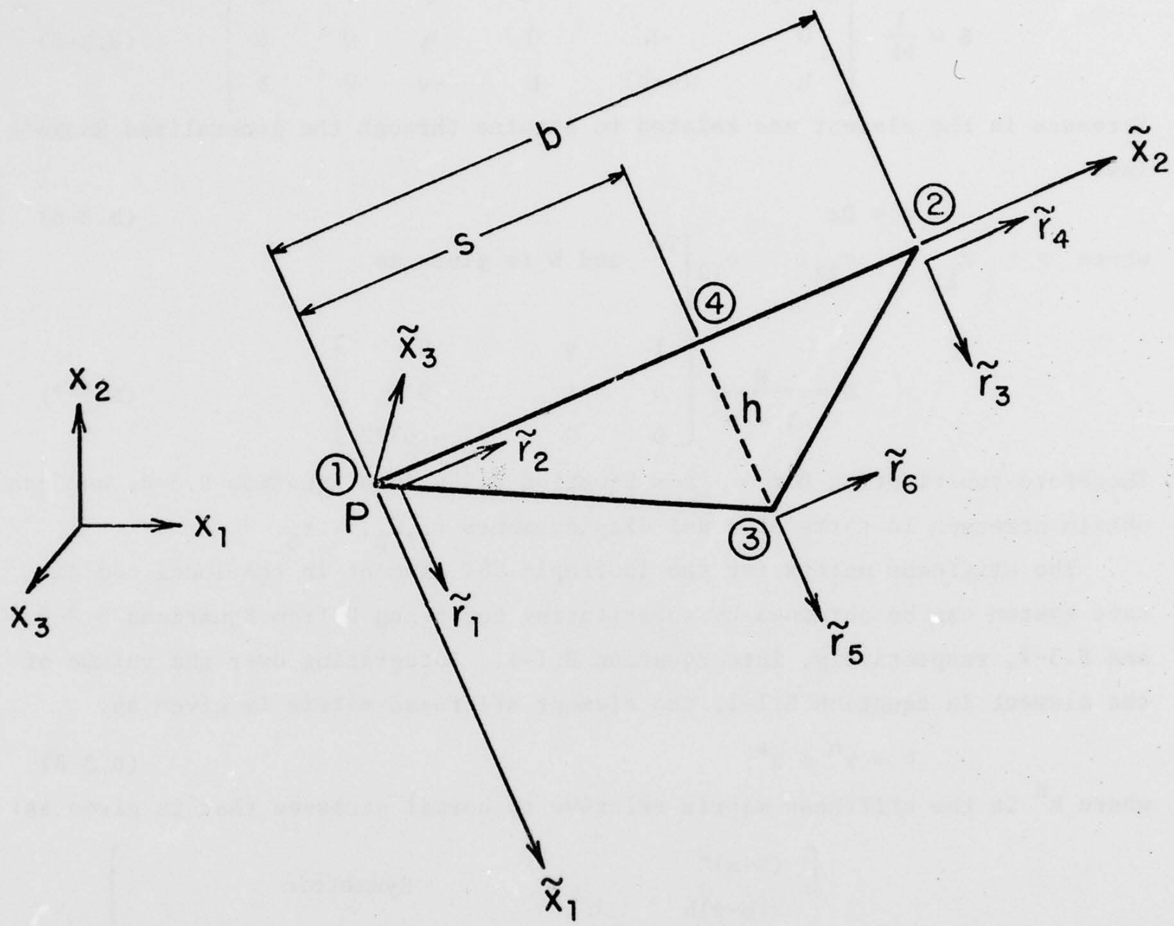


Figure B.2. Isotropic Constant Strain Triangular (CST) Element

Substituting Equation B.3-3 into Equation B.3-4, one can identify the matrix B as:

$$B = \frac{1}{bh}\begin{bmatrix} (s-b) & 0 & -s & 0 & b & 0 \\ 0 & -h & 0 & h & 0 & 0 \\ h & (s-b) & h & -s & 0 & b \end{bmatrix} \qquad (B.3\text{-}5)$$

Stresses in the element are related to strains through the generalized Hooke's law:

$$\sigma = D\varepsilon \qquad (B.3\text{-}6)$$

where $\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{12} \end{bmatrix}^T$ and D is given as

$$D = \frac{E}{1 - \nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \qquad (B.3\text{-}7)$$

Therefore substituting for ε from Equation B.3-4 into Equation B.3-6, one can obtain stresses in terms of nodal displacements $\tilde{r}_1, \tilde{r}_2, \ldots \tilde{r}_6$.

The stiffness matrix for the isotropic CST element in the local coordinate system can be obtained by substituting for B and D from Equations B.3-5 and B.5-7, respectively, into Equation B.1-1. Integrating over the volume of the element in Equation B.1-1, the element stiffness matris is given as:

$$k = k^n + k^s \qquad (B.3\text{-}8)$$

where k^n is the stiffness matrix relative to normal stresses that is given as:

$$\tilde{k}^n = \frac{Et}{2bh(1-\nu^2)}\begin{bmatrix} (b-s)^2 & & & & \text{Symmetric} & \\ \nu(b-s)h & h^2 & & & & \\ (b-s)s & \nu hs & s^2 & & & \\ -\nu h(b-s) & -h^2 & -\nu hs & h^2 & & \\ -(b-s)b & -\nu hb & -sb & \nu hb & b^2 & \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (B.3\text{-}9)$$

and $\tilde{k}^s$ is the stiffness matrix relative to shear stress that is given as:

$$\tilde{k}^s = \frac{Et}{4bh(1+\nu)}\begin{bmatrix} h^2 & & & \text{Symmetric} & & \\ (b-s)h & (b-s)^2 & & & & \\ -h^2 & -(b-s)h & h^2 & & & \\ hs & (b-s)s & -hs & s^2 & & \\ 0 & 0 & 0 & 0 & 0 & \\ -hb & -(b-s)b & hb & -sb & 0 & b^2 \end{bmatrix} \qquad (B.3\text{-}10)$$

The mass matrix for the isotropic CST element in the local coordinate system is obtained by substituting for N from Equation B.3-3 into Equation B.1-3. Carrying out the indicated integration, one obtains:

$$\tilde{m} = \begin{bmatrix} m^* & 0 & 0 \\ 0 & m^* & 0 \\ 0 & 0 & m^* \end{bmatrix} \tag{B.3-11}$$

where m* is given as:

$$m^* = \frac{\rho At}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{B.3-12}$$

In order to assemble stiffness and mass matrices for the entire structure, one needs to transform the element stiffness and mass matrices relative to a global coordinate system. It can be shown [4] that under any rotation of the coordinate system, the element mass matrix is invariant, that is $m=\tilde{m}$. In order to transform the element stiffness matrix relative to a global coordinate system, one needs to define a matrix β for the CST element and then use Equation B.1-2 to obtain k. The matrix β is given as:

$$\beta = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \tag{B.3-13}$$

where matrix R is given as:

$$R = \begin{bmatrix} \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \end{bmatrix} \tag{B.3-13}$$

where ℓ_1, m_1, n_1 and ℓ_2, m_2, n_2 are direction cosines of the x_1 axis (that is, the line 4-1) and the x_2 axis (that is, the line 1-2) relative to a global coordinate system x_1,x_2 and x_3. These direction cosines are given as:

$$\left.\begin{aligned} \ell_2 &= \frac{x_1^2 - x_1^1}{b}, \quad m_2 = \frac{x_2^2 - x_2^1}{b}, \quad n_2 = \frac{x_3^2 - x_3^1}{b} \\ \ell_1 &= \frac{(x_1^3 - x_1^1) - s\ell_2}{h}, \quad m_1 = \frac{(x_2^3 - x_2^1) - sm_2}{h}, \quad n_1 = \frac{(x_3^3 - x_3^1) - sn_2}{h} \end{aligned}\right\} \tag{B.3-14}$$

B.4. Symmetric Shear Panel Element (SSP)

In deriving the stiffness matrix for SSP elements (Figure B.3), the basic assumptions made are: 1) isotropic material, 2) uniform thickness, 3) rectangular configuration; if not rectangular, an equivalent rectangular plate of the same area is considered, 4) symmetric with respect to the middle surface, 5) plane stress state, 6) the stress distribution is assumed as follows:

$$\left.\begin{aligned} \sigma_{11}(\tilde{x}_1,\tilde{x}_2) &= \alpha_1\tilde{x}_2 + \alpha_2 \\ \sigma_{22}(\tilde{x}_1,\tilde{x}_2) &= 0.0 \\ \sigma_{12}(\tilde{x}_1,\tilde{x}_2) &= \alpha_3 \end{aligned}\right\} \tag{B.4-1}$$

where $\alpha_1,\alpha_2,\alpha_3$ are constants, and 7) the displacement boundary conditions are:

$$\left.\begin{aligned} \tilde{u}(0,b/2) &= \tilde{r}_1 \\ \tilde{u}(a,b/2) &= \tilde{r}_3 \\ \tilde{v}(0,b/2) &= \tilde{r}_2 \\ \tilde{v}(a,b/2) &= \tilde{r}_4 \\ \text{and} \quad & \\ \tilde{u}(\tilde{x}_1,0) &= 0.0 \end{aligned}\right\} \tag{B.4-2}$$

The local to global coordinate transformation for nodal displacements is expressed as:

$$\tilde{r} = \beta r \tag{B.4-3}$$

where

$$\beta = \begin{bmatrix} \ell_1 & m_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ell_1 & m_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.4-4}$$

$$r = \{\tilde{r}_1 \;\; \tilde{r}_2 \;\; \tilde{r}_3 \;\; \tilde{r}_4\}^T \tag{B.4-5}$$

and

$$r = \{r_1 \;\; r_2 \;\; r_3 \;\; r_4 \;\; r_5 \;\; r_6\}^T \tag{B.4-6}$$

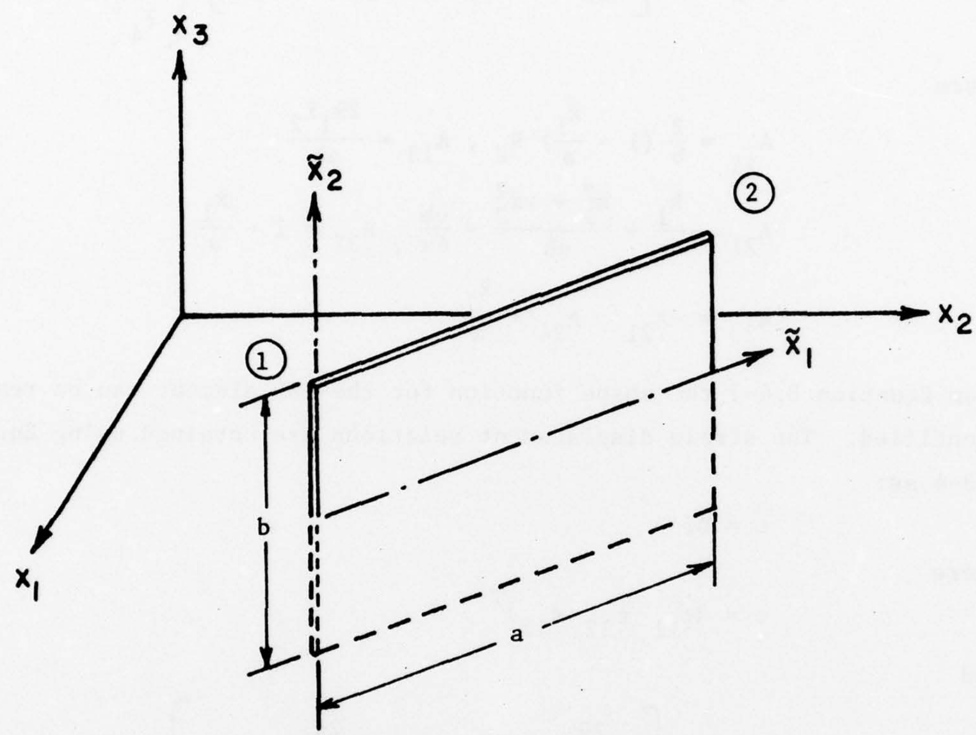


Figure B.3. Symmetric Shear Panel, or Symmetric Pure Shear Panel

From the assumed stress state, the strain-displacement and the boundary conditions, the displacement state can be obtained as:

$$\begin{Bmatrix} \tilde{u}(\tilde{x}_1,\tilde{x}_2) \\ \tilde{v}(\tilde{x}_1,\tilde{x}_2) \end{Bmatrix} = \begin{bmatrix} A_{11} & 0 & A_{13} & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} \end{bmatrix} \begin{Bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \tilde{r}_3 \\ \tilde{r}_4 \end{Bmatrix} \tag{B.4-7}$$

where

$$A_{11} = \frac{2}{b}\left(1 - \frac{\tilde{x}_1}{a}\right)\tilde{x}_2 \;,\; A_{13} = \frac{2\tilde{x}_1\tilde{x}_2}{ab}$$

$$A_{21} = -\frac{\tilde{x}_1}{b} + \frac{\tilde{x}_1^2 + \nu\tilde{x}_2^2}{ab} - \frac{\nu b}{4a} \;,\; A_{22} = 1 - \frac{\tilde{x}_1}{a} \tag{B.4-8}$$

$$A_{23} = -A_{21} \;,\; A_{24} = \frac{\tilde{x}_1}{a}$$

From Equation B.4-7 the shape function for the SSP element can be readily identified. The strain displacement relations are obtained using Equation B.3-4 as:

$$\varepsilon = B\tilde{r} \tag{B.4-9}$$

where

$$\varepsilon = \{\varepsilon_{11} \;\; \varepsilon_{22} \;\; \varepsilon_{12}\}^T \tag{B.4.10}$$

and

$$\beta = \begin{bmatrix} -\frac{2\tilde{x}_2}{ab} & 0 & \frac{2\tilde{x}_2}{ab} & 0 \\ \frac{2\nu\tilde{x}_2}{ab} & 0 & -\frac{2\nu\tilde{x}_2}{ab} & 0 \\ \frac{1}{b} & -\frac{1}{a} & \frac{1}{b} & \frac{1}{a} \end{bmatrix} \tag{B.4-11}$$

The stress-strain relation for plane stress is given by the generalized Hooke's law of Equation B.3-6. The matrix D is given in Equation B.3-7. Substituting in Equation B.3-6 the values of strains in terms of displacements, one obtains the stress-displacement relation as:

$$\sigma = \tilde{C}\tilde{r} \tag{B.4-12}$$

where

$$\tilde{C} = E \begin{bmatrix} -\frac{2\tilde{x}_2}{ab} & 0 & \frac{2\tilde{x}_2}{ab} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2(1+\nu)b} & \frac{-1}{2(1+\nu)a} & \frac{1}{2(1+\nu)b} & \frac{1}{2(1+\nu)a} \end{bmatrix} \qquad \text{(B.4-13)}$$

In the global coordinate system, the stress-displacement relation is:

$$\sigma = Cr \qquad \text{(B.4-14)}$$

where

$$C = \tilde{C}\beta \qquad \text{(B.4-15)}$$

The element stiffness matrix in the local coordinate system is:

$$k = \int_V B^T DBdV = t\int_s B^T DBds \qquad \text{(B.4-16)}$$

Thus, one obtains:

$$\tilde{k} = \frac{Et}{12(1+\nu)} \begin{bmatrix} \frac{2(1+\nu)}{\Theta} + 3 & -3 & -\frac{2(1+\nu)}{\Theta} + 3 & 3 \\ -3 & \frac{3}{\Theta} & -3 & -\frac{3}{\Theta} \\ -\frac{2(1+\nu)}{\Theta} + 3\Theta & -3 & \frac{2(1+\nu)}{\Theta} + 3\Theta & 3 \\ 3 & -\frac{3}{\Theta} & 3 & \frac{3}{\Theta} \end{bmatrix} \qquad \text{(B.4-17)}$$

Finally, the Von Mises equivalent stress σ^c for this element is given as:

$$\sigma^c = (\sigma_{11}^2 + 3\sigma_{12}^2)^{\frac{1}{2}} \qquad \text{(B.4-18)}$$

For calculating the maximum value of σ^c from Equation B.4-18 the following expressions for σ_{11} and σ_{12} are used (from Equation B.4-12):

$$\sigma_{11} = \frac{E}{a}(\tilde{r}_3 - \tilde{r}_1) \qquad \text{(B.4-19)}$$

and

$$\sigma_{12} = \frac{E}{2(1+\nu)} \left\{\frac{1}{a}(\tilde{r}_4 - \tilde{r}_2) + \frac{1}{b}(\tilde{r}_3 + \tilde{r}_1)\right\} \qquad \text{(B.4-20)}$$

The element mass matrix in the local coordinate system is obtained by substituting for N from Equation B.4-7 into Equation B.1-3. The elements of the symmetric (4x4) mass matrix are:

$$\tilde{m}_{11} = \frac{\rho b^2 t}{6}\left[\frac{\Theta}{3} + \frac{\nu\Theta}{6} + \frac{\Theta^3}{10} + \frac{\nu^2}{10\Theta}\right]$$

$$\tilde{m}_{12} = \frac{-\rho b^2 t}{24}\left[\Theta^2 + \nu\right] = \tilde{m}_{14}$$

$$\tilde{m}_{13} = \frac{\rho b^2 t}{6}\left[\frac{\Theta}{6} - \frac{\nu\Theta}{6} - \frac{\Theta^3}{10} - \frac{\nu^2}{10\Theta}\right] \qquad (B.4\text{-}21)$$

$$\tilde{m}_{22} = \frac{\rho b^2 \Theta t}{6}, \quad \tilde{m}_{23} = -\tilde{m}_{12}, \quad \tilde{m}_{24} = \tilde{m}_{22}/2$$

$$\tilde{m}_{33} = \tilde{m}_{11}, \quad \tilde{m}_{34} = -\tilde{m}_{14}, \quad \tilde{m}_{44} = \tilde{m}_{22}$$

B.5. Symmetric Pure Shear Panel (SPSP)

The element stiffness matrix for this pure shear element (Figure B.3) is also obtained by following the previous procedure and by assuming the stress state to be as follows: $\sigma_{11} = 0$, $\sigma_{22} = 0$, $\sigma_{12} = \alpha_1$, where α_1 is a constant. The element stiffness matrix is then given as:

$$\tilde{k} = \frac{Et}{4(1+\nu)}\begin{bmatrix} \Theta & -1 & \Theta & 1 \\ -1 & \frac{1}{\Theta} & -1 & -\frac{1}{\Theta} \\ \Theta & -1 & \Theta & 1 \\ 1 & -\frac{1}{\Theta} & 1 & \frac{1}{\Theta} \end{bmatrix} \qquad (B.5\text{-}1)$$

The stress state is

$$\sigma = \tilde{C}\tilde{r} \qquad (B.5\text{-}2)$$

$$\tilde{C} = \frac{E}{2(1+\nu)}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{b} & -\frac{1}{a} & \frac{1}{b} & \frac{1}{a} \end{bmatrix} \qquad (B.5\text{-}3)$$

APPENDIX C

to

Report Number 45

USER'S MANUAL FOR COMPUTER PROGRAMS

SOS4 AND DIMCO

C.1 Introduction

In this appendix, use of the computer program SOS4, (Structural Optimization by Substructures) for optimal design of structural and mechanical systems that can be idealized using Truss, CST and SSP elements, is described. The program is based on the algorithm and the logical sequence of computations of Chapters II, III and IV. It is developed in FORTRAN IV using the IBM 360-65 (370-158) computer at the University of Iowa. The program can be used for optimal design of structures with or without fail-safe constraints.

The program SOS4 has eighteen subroutines, namely VARI, ELESTF, STIFFM, RECALL, DECUPP, SOLDUP, MEVEC, DEFREQ, ZBZIEF, CONST, ABSMAX, GENC, DELBE, DESVV, SDD, SOLVEL, SUBSP and JACOBI. The subroutine VARI generates various variables for a substructure as shown in the statement COMMON/V2/ (see Appendix D). The subroutine ELESTF generates element stiffness matrix, element mass matrix and element stress matrix (required for the computation of bar forces in truss elements and stress components for CST and SSP/SPSP elements) for unit value of design variables. These quantities are stored in a vector form for subsequent use in design iterations. The subroutine STIFFM generates matrices $K_B^{(\alpha)}$ for the entire structure and $K_{II}^{(r,\alpha)}$ for each substructure. It used subroutine RECALL for generating element mass and stiffness matrices in the global coordinate system. It then uses subroutine DECUPP for decomposing upper band of matrices $K_{II}^{(r,\alpha)}$ (in case elements connecting interior nodes of the r^{th} substructure are damaged) and $K_B^{(\alpha)}$. The matrix $Q^{(r,\alpha)}$ is also computed in the subroutine STIFFM. The decomposed matrices $K_B^{(\alpha)}$ and $K_{II}^{(r,\alpha)}$ overwrite the original matrices.

The subroutine MEVEC is used to compute product of the structural mass matrix and the matrix of eigenvectors. Note that these calculations proceed elementwise. The subroutine DEFREQ computes sensitivity vector for a violated frequency constraint under all damage conditions. The subroutine ZBZIEF computes boundary displacements, interior displacements and element forces/stresses under all loading conditions. The subroutine CONST checks for the maximum stress under all loading conditions and previous damage conditions

for elements linked to a design variable. It also computes sensitivity vectors for violated stress constraints. The subroutine ABSMAX computes maximum nodal displacements under all loading conditions for a damaged structure. The maximum displacements are checked against their limit values and sensitivity vectors for violated constraints are computed. The subroutine GENC computes the matrix $C^{(\alpha)}$ of Equation 2.4-11.

The next three subroutines DELBE, DESVV, and SDD are used in computation of changes in design variables. Lagrange multipliers are computed and their signs are checked. Constraints corresponding to negative multipliers are taken out of the violated constraint set. The subroutine DESVV computes changes in design variables when only the design variable constraints are violated. The subroutine SOLVEL is based on the Gaussian elimination procedure and is used to compute the Lagrange multiplier vector μ. The last two subroutines SUBSP and JACOBI are used to compute the lowest eigenvalue and the corresponding eigenvector for each damage condition. These subroutines are based on the Subspace Iteration method coupled with the substructuring technique, as explained in Section 2.2.

A number of vectors and matrices are used in the main program as well as in the subroutines. In order to save computer storage, COMMON statements are used (see Appendix D). For each structure, dimensions of various matrices depend on the number of members, number of substructures, number of degrees of freedom, etc. Computation for dimensions of these matrices is explained later in this appendix. Once this information has been supplied, the computer program DIMCO (Dimension Computer; listed in Appendix D) can be used to generate and punch dimension cards for the main program and all its subroutines.

C.2. Data Organization

This section describes a procedure for setting up the problem and the input/output data organization for the computer program SOS4.

C.2.1. Problem Set-up

Setting up the problem is fairly simple. The complete structure, irrespective of the number of damage conditions, is divided into a number of substructures such that each substructure interacts with a minimum number of other substructures. A set of global axes for the structure is selected which is also used for each substructure. The numbering of nodes is done in two steps:

(i) Boundary nodes of each substructure are numbered first and then the interior nodes. The node numbers for each substructure begin with 1.

(ii) All the boundary nodes are also numbered in an overall system.

This numbering system simplifies many of the logical statements in the program. Hereafter, numbering of boundary nodes will imply numbering in the overall system.

C.2.2. Input Data

The input information required for the program is divided into four subsections:

(i) Input data common to all substrucrures

(ii) Input data for individual substructures

(iii) Input data for damaged structures

(iv) Other input data.

Variables of the program are defined and explained according to the READ statements appearing in the program (Appendix D). All the input information is supplied on regular computer cards.

C.2.2.1. Data Common to All Substructures:

1. NUNIT, NN, NSU, NDAM, NLC, NV, NCC, BNC, NBW, NPH, NSD, ISPSP - FORMAT (16I5).

   NUNIT = Code number for type of unit used; NUNIT = 0 for U.S. - British Units, and NUNIT = 1 for SI units.

NN = Code number for type of structure; NN = 2 for a 2D structure, and NN = 3 for a 3D structure.

NSU = Number of substructures.

NDAM = Number of damage conditions.

NLC = Number of loading conditions.

NV = Number of design variables.

NCC = Number of degrees of freedom.

BNC = Number of boundary degrees of freedom.

NBW = Upper bandwidth of boundary stiffness (K_B) matrix including the diagonal.

NPH = Expected size of the violated constraint set, that is, maximum number of constraints that may be violated in any design cycle.

NSD = Total number of expected stress, displacement and frequency constraint violations. Only NSD number of constraint violations can be corrected at any design cycle.

ISPSP = Code number for SPSP/SSP elements. If ISPSP = 0, the program SOS4 considers SSP elements, otherwise (ISPSP.NE.0) SPSP elements.

2. IFS, IDV, IFR, IBUK, IDIS, IBDIS, IPS, IPD, IPC, JUSTW, IAUTO - FORMAT (16I5)

IFS* = Number of iterations for which stress-ratio design is initially required.

IDV* = Code number for the frequency constraint.

IFR = If this variable is assigned a value of 1 and frequency constraint is to be imposed, then the program will correct only the frequency constraint in the first cycle.

IBUK* = Code number for buckling constraints.

IDIS* = Code number for interior displacement constraints.

IBDIS* = Code number for boundary displacement constraints.

IPS* = Code number for printing force or stress matrix at each iteration. When IPS = 1 force matrix will be printed, and when IPS = 2, the stress matrix will be printed.

IPD* = Code number for printing displacement matrix after each iteration.

IPC = Code number for printing stress and displacement constraint violations under each damaged condition.

JUSTW = Either 0 or 1:

If IDV = 0 and JUSTW = 0, then the program skips frequency analysis and design sensitivity analysis of the frequency constraint.

If IDV = 0 and JUSTW = 1, then the program calculates and prints the eigensolution. However the frequency constraint is not imposed.

If IDV = 1, then the program solves the eigenvalue problem and imposes the frequency constraint regardless (independent) of the input value for JUSTW.

IAUTO =
- 0; implies that the user wants to supply the matrix of eigenvectors to be used in Subspace Iteration.
- 1; implies that the matrix of eigenvectors will be automatically generated in the computer program at the start of the Subspace Iteration.

(*): If value assigned to this code is 0, then the corresponding command will be ignored. For example, if IBUK = 0, then buckling constraints will be ignored.

3. ILIM, ITRS, LNSV, LCON, (ITY(I) = 1,3), IWMM - FORMAT (16I5)

ILIM = Limit on the number of iterations or design cycles. The program stops if convergence is not obtained within this specified limit on number of iterations.

ITRS = Number of times the step size is to be changed. A provision is made in the program SOS4 to change the step size to any desired fraction of the original value if the variation of the cost function remains within the specified limit for a specified number of design cycles. This is done to obtain a finer convergence of the algorithm.

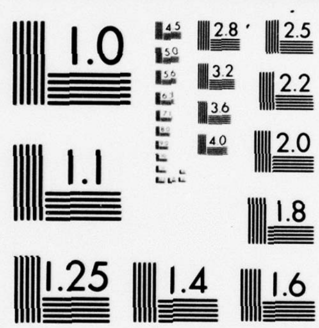
LNSV = Number of times the variation in the cost function should remain within the specified limit before the step size can be changed to any fraction of the original value.

AD-A065 936 IOWA UNIV IOWA CITY DIV OF MATERIALS ENGINEERING F/G 1/3
FAIL-SAFE OPTIMAL DESIGN OF STRUCTURES WITH SUBSTRUCTURING.(U)
AUG 78 D T NGUYEN, A K GOVIL, J S ARORA DAAK11-77-C-0023

UNCLASSIFIED TR-45 NL

2 OF 2
AD
A065936

END
DATE
FILMED
5-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ITY(1) = 4 if plane (6 if space) truss elements are present; otherwise 0.

ITY(2) = 9 if CST elements are present; otherwise 0.

ITY(3) = 6 if SSP elements are present; otherwise 0.

IWMM = $\begin{cases} 0, \text{ generates weighting matrix (see Chapter 4)} \\ 1, \text{ sets weighting matrix equal to identity matrix.} \end{cases}$

4. DF, RIT, RIN, RL, EP, STP1, STP2 - FORMAT (8F10.4).

DF = Requested reduction in the cost function for calculating the step size. This reduction factor is used in the regular computational algorithm and may be changed after some design cycles based on the criteria described above. For a five percent reduction in cost function, DF is assigned a value of 0.05. This variable may also be assigned 0 value, and in that case the program will correct only the violated constraints. The objective function will not be reduced.

RIT = Requested reduction in cost function for calculating the step size whenever all constraints are satisfied and ILIM > 0. This variable is used for a finer convergence near the optimum. If the regular step size is to be used then RIT = DF.

RIN = Requested reduction in the cost function for calculating a step size if all constraints are satisfied initially (RIN > 0). A larger step size may be taken if all the constraints are satisfied initially in order to speed up the convergence. For example, RIN = 0.25, if a 25 percent reduction in cost function is desired initially.

RL = Specified variation in the cost function for reducing step size, that is, if the variation in cost function should remain within one percent for two design cycles before the step size may be changed, then RL = 0.01 and LNSV = 2.

EP = A small number for checking ε-active constraints. A value of 0.02 to 0.0001 (2% to 0.01%) has been used in many calculations.

STP1 = A positive multiplier for changing DF and RIT (see LNSV in card no. 3).

STP2 = A positive multiplier for changing RL (see LNSV in card no. 3).

5. (FACC(I), I = 1,3), RF, CONL, FORMAT (8F10.3).

FACC(I)*= Multiplier associated with weighting matrix (refer to Ch. 4).

FACC(1): for truss elements

FACC(2): for CST elements

FACC(3): for SSP/SPSP elements

RF = Resonant frequency for the truss in cycles per second (Hertz). When IDV > 0, RF cannot be zero.

CONL = Maximum constraint violations to be corrected. This parameter is always negative. If any constraint violation is smaller than this amount, only this amount will be corrected. For example, CONL = -1.0 implies $\Delta\tilde{\phi}$ = -1.0 for any $\Delta\phi$ < -1.0. Generally, a large value is used for this parameter; a value of -100 is recommended.

6. ERR1, ERR2, ERR3, ERR4, ERR5 - FORMAT(5E16.7)

ERR1 = Error criteria used for checking convergence of eigenvalues in the Subspace Iteration method. A value of 0.100E-05 for ERR1 has been used quite often in computation.

ERR2 = Tolerance in design variables in percent at the optimum. At each design cycle, the percent change in each component of the design variable vector is checked and if each component is within ERR2, then the design variable vector is assumed to have converged. The value assigned to ERR2 is 0.100E-02 if a convergence of 0.1 percent is sought.

ERR3 = Constraint violation telerance in percent at the optimum point. The value assigned to ERR3 is 0.100E-2 if, at the optimum point, each violation of a constraint is to be within 0.1 percent.

ERR4 = Tolerance in the cost function in percent at the optimum. The value assigned to ERR4 is 0.100E-02 if, at the optimum point, the cost function variation is to be within 0.1 percent. If all the convergence criteria, that is, ERR2, ERR3, and ERR4 are satisfied then the convergence to the optimum is assumed and the design process is stopped.

---

* to be selected by the designer

ERR5 = Error criterion used in checking zero elements in Gaussian elimination procedure. A value of 0.100E-05 has been used in the present computations.

7. (DLIB(I), I = 1, BNC) - FORMAT (8F10.3)

The boundary displacements limits for the structure in inches (metres) are supplied in this statement. The total number of cards for this step depends upon the value of BNC because each card contains only eight numbers. These displacement limits are punched in a definite order determined by the order of numbering the boundary joints of the structure. For example, if joint number 1 has all three degrees of freedom then it will have displacement numbers 1, 2, and 3; if joint 2 has two degrees of freedom then displacement numbers 4 and 5 will be for these two degrees of freedom, and so on.

8-10. This set of input data cared contains information about the loaded boundary nodes only. The boundary load matrix of dimension (BNC x NLC), is initialized first and then for each loading condition, following information is READ according to the specified format.

8. First card contains NLJ, the number of loaded boundary nodes; FORMAT (16I5).

9. The next set of cards contains node numbers of loaded joints in the overall boundary node numbering syste The number of cards depends on NLJ as each card contains only sixteen numbers; FORMAT (16I5).

10. The last set of information, punched on separate cards, contains the node number and loads in kips (Newton) applied along permissible degrees of freedon; FORMAT (I5, 3F10.2).

11-12. This set of cards provides information about design variable linking of members across the substructure boundaries.

11. LINK - FORMAT (I5)

LINK = Number of design variables linking across substructure boundaries.

12.* LINLG(I,1), LINLG(I,2); I = 1, LINK; FORMAT (16I5).

LINLG(I,1) = Type of element

LINLG(I,2) = Design variable group to which the element is linked.

* If LINK = 0, skip #12.

C.2.2.2. Data for Individual Substructures: In this section of the program input data for each substructure is READ separately in a proper sequence. The total number of such sets of data is equal to NSU. The following input information is given for the r^{th} substructure:

13. NJ(r), NBJ(r), NCB(r), NIC(r), NBW1(r), NBW2(r), NBW3(r) - FORMAT (16I5).

    NJ(r) = Total number of nodes.

    NBJ(r) = Number of boundary nodes.

    NCB(r)* = Number of boundary degrees of freedom.

    NIC(r)* = Number of interior degrees of freedom

    NBW1(r)* = Upper bandwidth of the matrix $K^{(r)}$ including the diagonal.

    NBW2(r)* = Upper bandwidth of the matrix $K_{BB}^{(r)}$ including the diagonal.

    NBW3(r)* = Upper bandwidth of the matrix $K_{II}^{(r)}$ including the diagonal.

* These parameters for the stiffness matrix are explained in Figure C.1.

14. NZ(I,K); I = 1,NB - FORMAT (16I5); NB = NBJ(r).

    This set of data cards contains information about interconnection between boundary nodes in the overall and the substructural numbering systems. The number of boundary nodes for the r^{th} substructure is NBJ(r), and they are numbered in an ascending order starting from 1. In the overall boundary numbering system, these NBJ(r) nodes will correspond to some boundary nodes in the overall system. For example, if r^{th} substructure has 5 boundary nodes, then they will be numbered 1, 2, 3, 4 and 5 in the substructural or local boundary node numbering system. In the overall system, let these nodes correspond to nodes 10, 11, 12, 13 and 14. Then for this data set, the number 10, 11, 12, 13 and 14 will be punched according to above format.

15. J, X(J,r), Y(J,r), Z(J,r), (ND(I), I = 1, NN) - FORMAT (I5, 3F10.3,3I5).

    J = Nodal number

    X(J,r), Y(J,r), Z(J,r) = X, Y, Z (or x_1, x_2, x_3) coordinates of the J^{th} node in the global Cartesian coordinate system. Units: inches (metres).

    The remaining integers are the code numbers for this node. Each node has its degrees of freedom, that is, displacements in coordinate directions x_i, i = 1 to NN. If displacement along a particular coordinate axis is allowed then that code number is assigned a value of 1,

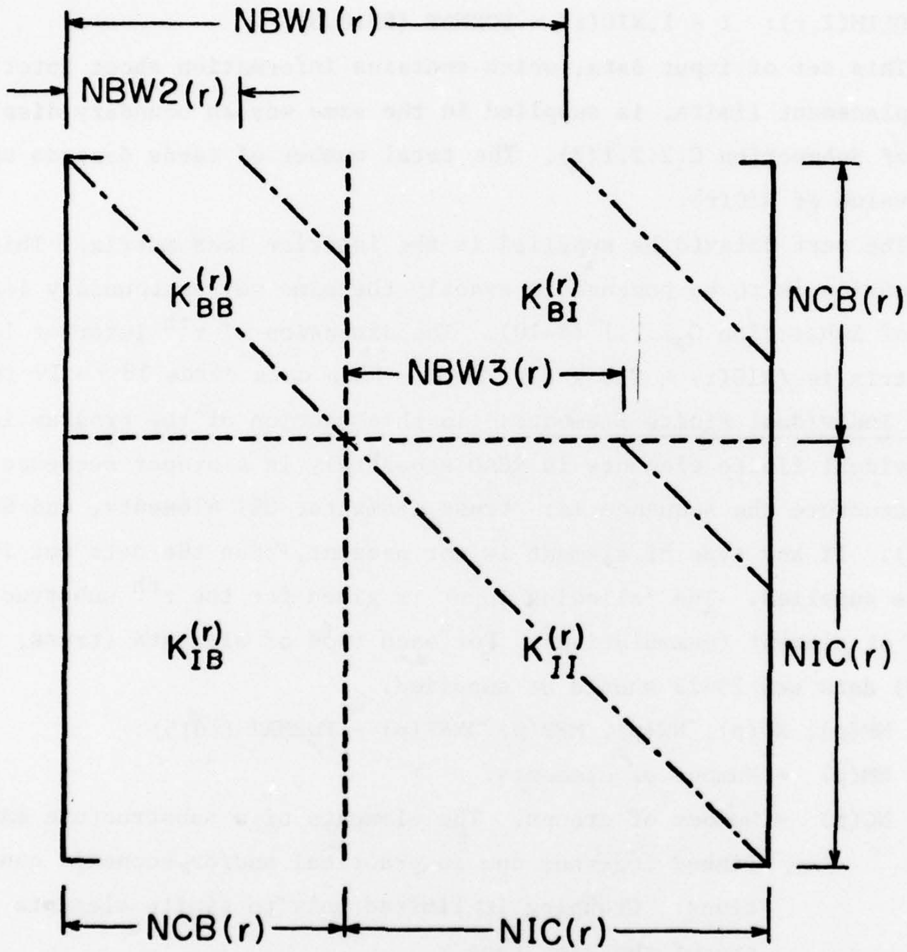


Figure C.1. Bandwidth Parameters for Stiffness Matrix of the r^th Substructure

otherwise it is zero. For example, the code number 1, 1, and 0 for a particular node specify that the displacement in the x_3-direction of this node is zero. The total number of cards for this step is NJ(r) and they must be placed in an ascending order. Note: Skip data card numbers 16 to 19 if NIC(K) = 0.

16. DLIM(I,r): I = 1,NIC(r) - FORMAT (8F10.3)

    This set of input data, which contains information about interior displacement limits, is supplied in the same way as boundary displacements of Subsection C.2.2.1(7). The total number of cards depends upon the value of NIC(r).

17-19. The next data to be supplied is the interior load matrix. This information is to be punched in exactly the same way as boundary load matrix of Subsection C.2.2.1 (8-10). The dimension of r^{th} interior load matrix is (NIC(r) x NLC x r). Note: Skip data cards 18 to 19 if NLJ = 0.

Data for Individual Finite Elements: In this section of the program input for individual finite elements is READ separately in a proper sequence (for r^{th} substructure the sequence is: truss elements, CST elements, and SSP/SPSP elements). If any type of element is not present, then the data set 20-23 is not to be supplied. The following input is given for the r^{th} substructure and p^{th} type of element (cummulative). For each type of elements (truss, CST, SSP/SPSP) data set 20-23 should be supplied.

20. NM(p), NG(p), NW(p), MEB(p), MEF(p) - FORMAT (16I5).

    NM(p) = Number of elements.

    NG(p) = Number of groups. The elements of a substructure may be linked together due to practical and/or economic considerations. Grouping is limited only to finite elements that are of the same type.

    NW(p) = Number of design variables (NW(r) $\leq$ NG(r)). If cross-sectional area of each member of r^{th} substructure is to be considered as a design variable, then NM(r) = NG(r) = NW(r). If NW(r) < NG(r), then only the first NW(r) groups are considered as design variables.

    MEB(p) = Number of the first element.

    MEF(p) = Number of the last element.

21. J, L, (MN(N+M, p), M = 1, L) - FORMAT (16I5)

(Initially N = 0 and later N = N + L.).

This set of cards contains information about grouping of elements. Information about each group starts on a new card. The number of cards for this step is NG(p) and are placed in an ascending order of group number.

J = Group number.

L = Number of elements in the J^{th} group.

(MN(N+M, p), M = 1, L) = element numbers of the J^{th} group.

22. BL(J,p), BU(J,p), ALP(J,p), SL(J,p), SU(I,p), RO(J), XNUU(J,p), E(J,p) - FORMAT (8F10.3).

This set of input data cards contains information about upper and lower bound and material properties for the elements of a group. The number of cards for this step is equal to NG(p) and they must also be placed in an ascending order of group numbers. Each card contains the following information about the elements of a group (say J^{th}).

BL(J,p)* = Lower limit on the design variable. It should be noted that this must be a non-zero positive number.

BU(J,p)* = Upper limit on the design variable.

ALP(J,p) = Constant $\bar{\alpha}_i$ for each truss element of the group. This is needed for computing the moment of inertia of an element, $I_i = \bar{\alpha}_i b_i^2$. For CST and SSP elements, any value may be used.

SL(J,p) = Compressive stress limit in kips per square inch (Newton/m^2); punched as a positive number.

SU(J,p) = Tensile stress limit in kips per square inch (Newton/m^2); punched as a positive number.

RO(J) = Specific weight of the material in pounds per cubic inch (Newton/m^2).

XNUU(J,p) = Poisson's ratio of the material.

E(J,p) = Modulus of elasticity of the material in kips per square inch (Newton/m^2).

*For truss elements: $inch^2$ ($metre^2$); for CST and SSP elements: inch (metre).

23. M8, JP, JQ, JR, MPC(M8,p) - FORMAT (16I5).

This set of input data cards contains information about the element connectivity. The number of cards for this step is equal to NM(p) and they must also be placed in an ascending order of elements. Each card contains the following information about the element:

M8 = Element number.

JP, JQ, JR = Element end nodes. For truss and SSP/SPSP elements, skip JR.

The last information on this card defines the type of element connection according to the following code:

M(M8,p) =
- -1, implies element connected to boundary nodes only.
- 0, implies element connected to both boundary and interior nodes.
- +1, implies element connected to interior nodes only.

C.2.2.3. Input Data For Damaged Structures: In this section of the program, input data for each damage condition is READ separately in a proper sequence (skip this section if NDAM = 0). The total number of such sets of data is equal to NDAM. The following input information is given for the I^{th} damage condition.

24. RRF(I), RDLIM(I), RSL(I), RSU(I), RLOAD(I) - FORMAT (8F10.3).

This set of cards contains values of multipliers to be used in defining the frequency limit, displacement limits, stress limits and applied load for the I^{th} damage condition. Each card contains the following information for the I^{th} damage condition:

RRF(I) = Multiplier for lower bound on natural frequency.

RDLIM(I) = Multiplier for admissible displacements.

RSL(I) = Multiplier for lower limit on stress (compressive stress).

RSU(I) = Multiplier for upper limit on stress (tensile stress).

RLOAD(I) = Multiplier for applied loads.

For example, RRF(2) = 0.75 implies that the resonant natural frequency under damage condition number 2 is three-fourths that of the undamaged structure.

The next four input data sets (25-28) for I^{th} damage condition are READ in the following order:

DO α r = 1, NSU

25. READ KIIDAM(r,I) - FORMAT (16I5)

DФ α p = 1, 3 (TRUSS, CST, SSP/SPSP)

IF (ITY(p).EQ.0) GO TO α (see #3)

26. READ N - FORMAT (16I5).

IF (N.EQ.0) GO TO α.

27. READ NDM(J), J= 1, N

28. READ REDUC(J), J = 1, N

α CONTINUE

Here input data set number 25 contains damage code for the matrix $K_{II}^{(r)}$ as follows:

$$\text{KIIDAM(r,I)} \begin{cases} = 0, \text{ implies that the matrix is not changed due to damage.} \\ \neq 0, \text{ implies the matrix is changed due to damage.} \end{cases}$$

In data set number 26, N is the number of elements damaged in the I^{th} damage condition. Note: Skip data set number 27 and number 28 if N = 0. Data set number 27 contains identification numbers for damaged elements. The number of cards depends upon the value of N, since each card contains at the most 16 values (FORMAT (16I5)).

NDM(J) = the J^{th} damaged member in the I^{th} damage condition.

For example, in damaged condition number 1 if there are 6 damaged members: 1, 4, 6, 71, 75 and 76, then:

$$\left.\begin{array}{l} \text{NDM(1)} = 1 \\ \text{NDM(2)} = 4 \\ \vdots \\ \text{NDM(6)} = 76 \end{array}\right\} \text{These 6 numbers can be punched on one data card (FORMAT (16I5)).}$$

In data set number 28, a reduction ratio for each damaged member is given to define the extent of damage. The number of cards depends upon the value of N since each card contains at the most 8 values (FORMAT (8F10.3)). A total loss of the member is denoted by specifying 1.0 to its reduction ratio. In the above example, if percentage of damage to members, 1, 4, 6, 71, 75 and 76 are 10%, 40%, 60%, 90%, 100% and 20%, respectively, then:

$$\left.\begin{array}{l} \text{REDUC(1)} = 0.100 \\ \text{REDUC(2)} = 0.400 \\ \vdots \\ \text{REDUC(6)} = 0.200 \end{array}\right\} \text{These numbers can be punched on one data card (FORMAT (8F10.3)).}$$

C.2.2.4. Other Input Data:

29. Skip this set of data if IDV = 0 and JUSTW = 0, or if IAUTO = 1. Otherwise, supply the matrix of eigenvectors according to the Format 5E16.7; XEIG(J,I) where J = 1,2...NCC and I = 1,2.

Note that in the Subspace Iteration, two eigenvectors are needed to accurately calculate the lowest eigenvalue. The input matrix of eigenvectors XEIG(J,I) need to be in the following form:

XEiG(J,I) =

BNC	1 2 ⋮ BNC	Total number of boundary DOF for the complete structure.
NIC(1)	BNC+1 ⋮ BNC+NIC(1)	Total number of interior DOF for substructure 1.
NIC(2)	BNC+NIC(1)+1 ⋮ BNC+NIC(1)+NIC(2)	Total number of interior DOF for substructure 2.
NIC(r^{th})	BNC+NIC(1)+NIC(2)+1 ⋮ NCC	NCC is the total number of DOF for the complete structure.

The last two input data sets (#30 and #31) are READ in the following order:

DO α r = 1, NSU

DO α p = 1, 3

IF (ITY(p).EQ.0) GO TO α (see #3)

30. READ B(I,p), I=1, NG(p) - FORMAT (8F10.3)

31. READ IGRT (I,P), I=1, NG(p) - FORMAT (16I5)

α CONTINUE

Input data set number 30 contains starting valued of design variables (cross-sectional area in inch^2 (metre^2) for truss elements, and thickness in inches (metres) for CST and SSP/SPSP elements) and must be placed in the ascending order of group numbers.

Input data set number 31 defines status of the design variable (DV) grouping.

$$\text{IGRT(I,p)} = \begin{cases} -1, & \text{implies that DV is linked to DV of previous sub-structure} \\ 0, & \text{implies that the DV is fixed} \\ +1, & \text{implies that the DV is free, that is, neither linked nor fixed.} \end{cases}$$

C.2.3. Output

Two types of outputs are received from the computer program; printed output and punched output on computer cards. In the printed output, all of the input data is first printed out for verification purposes. At each design cycle, value of the cost function, values of the design variables, type and number of constraint violations, and the member force matrix are printed out. Also, Lagrange Multipliers, changes in design variables and the cost function history are printed out.

The punched output, consisting of three sets of data cards, corresponds to the data required in set numbers, 29, 30 and 31, respectively. If IDV = 0, then the first data set, consisting of eigenvectors of last design cycle, is not punched. The last two data sets, consisting of design variables of last iteration and their status (linked, fixed or free) are punched out for subsequent computer runs, if necessary.

C.3. Computation of Dimensions of Various Matrices

The dimensions of various matrices and vectors depend upon the size of the structure considered. Various variables like BNC, NLC, NCI(K), etc. as defined in Section C.2, determine sizes of various matrices. For easy computation of dimensions, the dimension statements used in the program (Appendix D) are explained here in terms of these variables.

DIMENSION	PB(BNC,NLC), ALP(NGU,KKU), DBIN(ILIM,2), OO(NV), FACC(3), FB(ILIM), BETA(2*SN), CL(3), NZ(NBJL,NSU), LINLG(LINK,2), NJL(NLJ, NVV(3), NEGV(NDAM+1)
COMMON/V2/	NIC(NSU), NW(KKU), NG(KKU), NBW1(NSU), NBW2(NSU, NBW3(NSU), NM(KKU), NBJ(NSU), NJ(NSU), NCB(NSU), NEW(NSU), IQS(NSU), MEB(KKU), MEF(KKU)
COMMON/P1/	B1(9,9), B2(9,9), B3(9,9), ESF(9,9), NA(MAX(NM,9)), NI1(9), NJ1(9), NJ2(9)

COMMON/P2/ XNUU(NGU,KKU), ELL(M8,K21), BU(NGU,KKU), STRESS(NTE*3+NCE*27+NSE*12), TCSM(NTE+NCE+NSE*21), TRCSSP(NTE*6+NCE*45+NSE*21), XCOST(3), ICSS(M8,K21), ISAC(M8,K21), INDC(M8,K21), IGRT(NGU,KKU), IGRE(M8,K21), NNDC(NTE*6+NCE*9+NSE*6), LLN(3), ITY(3), ICSSM(M8,K21)

COMMON/P3/ EVEC(NCC,NDAM+1), RRF(NDAM+1), RDLIM(NDAM+1), RSL(NDAM+1), RSU(NDAM+1), RLOAD(NDAM+1), REDUC(K22), NDOF(NDAM+1), NDM(K22), NBDAM(KKU,NDAM), KIIDAM(NSU,NDAM+1)

COMMON/P4/ INF(NSD,8), NGV(NGU,KKU), INO(NSD), NDISP(NCC)

COMMON/P5/ YK(NCC), YM(NCC), SK(NCC), SM(NCC), EY(NCC), SG(NCC)

COMMON/R1/ BL(NGU,KKU), DLIB(BNC)

COMMON/R2/ PI(NCIL,NLC,NSU), RR(M8,K21), E(NGU,KKU), MN(M8,KKU), MON(NGU,KKU), MN(M8,KKU), NOM(NGU,KKU)

COMMON/R4/ IIL(NSD,NSU), KLC(NSD), IOK(NSU), NO(NLC)

COMMON/R5/ B(NGU,KKU), SL(NGU,KKU), SU(NGU,KKU), DPB(K1, K2), DLIM(NCIL,NSU), SS(NV)

COMMON/A1/ Q(NCIL,NCBL,NSU), ZI(NCIL,NLC,NSU), C(BNC,NBW), SB(BNC,NLC)

COMMON/A3/ BR(M8,K21), TRSF(NTE,NLC), CSTF(NCE,NLC,4), SSPF(NSE,NLC,3), Z(NV,NSU), SZE(NPH), MP(M8,K21), ND(K3)

COMMON/A4/ X(maxo(NJ(r),NTE),NSU), DLP(NPH), DLPH(NPH, T(K4), WM(NV, RO(NV)

COMMON/A5/ D(K5,K6), DS(K5,K7), A2(BNC,K9), DKI(NCI,NU3), KIIUBW(NSU)

COMMON/A6/ DPZ(K10,K9), ZZ(K11,K12), BE(K11,K12), W(K11), H(K26), VV(K13), Y(M8,NSU), NZC(NCBL,NSU)

COMMON/A7/ DPX(NGG,NSD)

COMMON/C1/ XEIG(NCC,2), YXEIG(NCC,2), WS(2), DM(1,1), IET(NDAM+1)

COMMON/C3/ QQK(2,2), QQM(2,2), QA(2,2)

COMMON/C4/ ETC(NV*IPDAM), TEI(IPDAM), TE(IPDAM)

where

$$\left.\begin{aligned} \text{NBJL} &= \max_r \{\text{NBJ}(r)\} \\ \text{NCIL} &= \max_r \{\text{NIC}(r)\} \\ \text{NCBL} &= \max_r \{\text{NCB}(r)\} \end{aligned}\right\} \quad r = 1 \text{ to NSU}$$

$$\text{NU3} = \sum_{r=1}^{\text{NSU}} \text{NBW3}(r)$$

NGU = maximum number of groups for any type of finite element in a substructure

KKU = NSU*K21

NLJ = number of loaded nodes.

M8 = maximum of truss, CST or SSP/SPSP elements in the structure.

= max(NTE,NCE,NSE)

NTE = number of truss elements

NCE = number of CST elements

NSE = number of SSP/SPSP elements

NM = NTE+NCE+NSE

NGG $= \sum_{k=1}^{NSU} NG(k)$

IPDAM = NDAM+1

PN = 2*SN , SN = 2*NN

K1 = max(NV,NCIL) , K2 = max(NSD,NU3)

K3 $= \max\ NPH,\ SN^* \sum_{I=1}^{NSU} NJ(I)$

K4 = max NPH,NM

K5 = max(NCIL,BNC , K6 = max(NU3,NBW)

K7 = max(NSD,NCBL + NLC) , K9 = max(NSD,NCBL)

K10 = max(NSD,NCIL) , K11 = max(NPH,NV)

K12 = max(NLC,3) K13 = max(NPH,NM)

K21 = number of finite elements used

K22 = total number of damaged members under all damage conditions

K26 = max(NV,M8)

After dimensions of various matrices have been determined, the computer core requirements can easily be specified. For IBM 360/65, the compilation step in double precision, requires a computer core of 184K, regardless of dimensions of various matrices.

C.4. User's Manual for the Computer Program DIMCO

As noted earlier, the computer program SOS4 has eighteen subroutines. Each subroutine has several COMMON statements. These statements are dependent on a structural design problem. It is cumbersome and time consuming to punch these cards for each structural design problem. Therefore, a computer program DIMCO (Dimension Computer) has been developed to calculate dimensions of various

matrices and to generate COMMON statements for all subroutines of SOS4. For each structural design problem, the program DIMCO can be used to generate dimension cards for the program SOS4 and each of its subroutines.

The program DIMCO requires only a few simple input data cards (in integer FORMAT) as described below:

<u>Card #1</u> (FORMAT 16I5)

$$\text{NN} = \begin{cases} 2, & \text{for 2 dimensional structure} \\ 3, & \text{for 3 dimensional structure} \end{cases}$$

NSU = number of substructures

NDAM = number of damage conditions

NLC = number of loading conditions

NV = number of design variables

NCC = total number of degrees of freedom (DOF)

BNC = total number of boundary DOF

NBW = upper banwidth of the matrix K_B (Effective boundary stiffness matrix)

NPH = maximum number of constraint violations allowed at any design iteration

NSD = maximum number of stress, displacement and natural frequency constraint violations to be corrected at any design iteration

ITE = number of different type of elements for the structure

NBLJ = number of boundary loaded joints for the undamaged structure

NDMT = total number of damaged members

$$\text{LINK} = \begin{cases} 0, & \text{when there is no design variable linking with previous substructures} \\ 1, & \text{when there is (are) design variable(s) linking to previous substructures} \end{cases}$$

ILIM = maximum number of design iterations allowed

<u>Card #2</u> (FORMAT 16I5)

ITY(1) = 1 if truss elements exist; 0 otherwise

ITY(2) = 1 if CST elements exist; 0 otherwise

ITY(3) = 1 if SSP/SPSP elements exist; 0 otherwise

<u>Data Set #3</u> (also refer to Figure C.1 of Appendix C)

(i) Information about the K^{th} substructure where K=1,2...,NSU (FORMAT 16I5)

NJ(K) = number of joints for the K^{th} substructure

NBJ(K) = number of boundary joints for the K^{th} substructure

NCB(K) = number of boundary DOF for the K^{th} substructure

NIC(K) = number of interior DOF of boundary joints for the K^{th} substructure

NBW1(K) = upper bandwidth of the entire stiffness matrix for the K^{th} substructure

NBW2(K) = upper bandwidth of the matrix K_{BB} for the K^{th} substructure

NBW3(K) = upper bandwidth of the matrix K_{II} for the K^{th} substructure

NILJ(K) = number of interior loaded joints for the K^{th} substructure

(ii) Information about the J^{th} type of elements in the K^{th} substructure where J=1,2,3 (FORMAT 16I5). Omit this data set if ITY(J) = 0.

NM(KK) = number of J^{th} type of elements for the K^{th} substructure

NG(KK) = number of groups for the J^{th} type of elements and the K^{th} substructure

NW(KK) = number of design variables for the J^{th} type of elements and the K^{th} substructure

MEB(KK) = beginning member number of the J^{th} type of elements for the K^{th} substructure

MEF(KK) = final member number of the J^{th} type of elements for the K^{th} substructure

For an open truss helicopter tail boom with 3 substructures and 1 element type (truss), K=3 and ITE=1. Therefore a total of only 8 input cards (1+1+6) are required. In general, a total of p cards are required for the computer program DIMCO where p = 2 + (NSU) * (1 + ITE).

APPENDIX D

to

Report Number 45

LISTING OF PROGRAMS SOS4 AND DIMCO

D.1. Listing of the Program SOS4

```
/FSSOS JOB (--------,30,30,2001),'DI NTDUC',TIME=25                    JOB 603
*MESSAGEL     PLEASE INTERPRETE MY OUTPUT PUNCHED CARDS
/ EXEC FORTCLG,REGION=450K,TIME=25
/FORT.SYSIN DD *
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      DIMENSION PB( 36, 1),ALP( 14, 6),DBIN( 20, 2),OD( 51),FACC( 3),FB(
     1 20),BETA(12),CL( 3),NZ(  8, 3),LINLS( 1, 2),NJL( 4),NVV( 3),NEGV(
     2 7)
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/V2/NIC( 3),NW( 6),NS( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCE( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),VA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),ISRT( 14, 6),IORE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P3/EVEC(  1, 1),RRF( 7),RDLIM( 7),RSL( 7),RSU( 7),RLOAD( 7)
     1,REDUC( 90),NDOF( 7),NDM( 90),NBDAM( 6, 6),KIIDAM( 3, 7)
      COMMON/P4/INF( 50, 8),NGV( 14, 6),INO( 50),NDISP( 72)
      COMMON/P5/YK(  1),YM(  1),SK(  1),SM(  1),EY(  1),SG(  1)
      COMMON/R1/BL( 14, 6),DLIB( 36)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/R4/IIL( 50, 3),KLC( 50),IOK( 3),NO( 1)
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
      COMMON/A7/DPX( 62, 50)
      COMMON/C1/XEIG( 72, 2),YXEIG( 72, 2),WS( 2),DM(  1,  1),IET( 7)
     $/C3/ QQK( 2, 2),QQM( 2, 2),QA( 2, 2)
      COMMON/C4/ETC( 357),TEI( 7),TE( 7)
C*********************************************************************
C*                                                                   *
C* PROGRAM    - 'FSSOS' FAIL-SAFE STRUCTURAL OPTIMIZATION WITH       *
C*                       SUBSTRUCTURING                              *
C* PROGRAMMER - ASHOK K. GOVIL                                       *
C* DIVISION OF MATERIALS ENGINEERING,                                *
C* UNIVERSITY OF IOWA, IOWA CITY, IOWA 52240                         *
C* AUGUST, 1977                                                      *
C*                                                                   *
C* FAIL-SAFE OPTIMAL DESIGN OF FINITE DIMENSIONAL MECHANICAL SYSTEMS*
C* SUBJECTED TO STATIC LOADING                                       *
```

```
C     * CONSTRAINTS ON - DIRECT STRESS/VON MISES EQUIVALENT STRESS, NODAL*
C     *                  DISPLACEMENT, FREQUENCY, AND BOUNDS ON DESIGN    *
C     *                  VARIABLES                                        *
C     *                                                                   *
C     * SUBSTRUCTURE FORMULATION IS USED                                  *
C     * STIFFNESS MATRIX METHOD IS USED TO ANALYZE THE STRUCTURE          *
C     * VARIOUS MEMBER OF THE STRUCUCTURE MAY BE GROUPED TOGATHER         *
C     * FINITE ELEMENT LIBRARY INCLUDES TRUSS, CST, AND SSP/SPSP ELEMENTS*
C     * ALL CALCULATIONS ARE IN DOUBLE PRECISION                          *
C     *********************************************************************
C     * UPDATED (JUNE 1978)NGUYEN THAI DUC                                *
C     * UNDER SUPERVISION OF PROF. J. S. ARORA                            *
C     * OPTION OF USING SUBSPACE ITERATION TO SOLVE EIGEN PROBLEM         *
C     * VIOLATIONS IN FREQ. & DISPL. OF EACH DAMAGED CONDITION ARE        *
C     * INCLUDED IN THE VIOLATED CONSTRAINT SET                           *
C     *********************************************************************
C       FORMATS-READ STATEMENTS
    4 FORMAT(8F10.6)
    5 FORMAT(8F10.3)
    8 FORMAT(5E16.7)
    9 FORMAT(I5,3F10.4,6I5)
   10 FORMAT(16I5)
   11 FORMAT(7F10.4,F10.2)
C       FORMAT-WRITE STATEMENTS
   12 FORMAT(3X,'SOME ERROR IN KC')
   19 FORMAT(////30X,'** DEPENDENT STIFFNESS MATRIX **  N=',I5,',   K=',I
     *2,',   IDC=',I2)
   24 FORMAT('1','  DATA COMMON TO ALL SUBSTRUCTURES ')
   25 FORMAT(//1X,'SN   STRUCTURE NUMBER =',I4/1X,'NSU   NO. OF SUBSTRUCTU
     1RES =',I4/1X,'BNC   OVERALL BOUND. DEGREES OF FREEDOM =',I4/1X,'NBW
     2  OVERALL  BOUND. UPPER BAND WIDTH =',I4/1X,'NLC   NO. OF LOADING C
     3ONDITIONS =',I4/1X,'NPH   TOTAL NO. OF EXPECTED CONSTR. VIOLATIONS
     4=',I4/1X,'NSD   NO. OF STRESS & DISPL. CONSTR. VIOLATIONS =',I4)
   26 FORMAT(/1X,    'IBUK=0 WILL NOT CONSIDER BUCKLING CONSTR =', I5/1X,
     1   'IDIS=0 WILL NOT CONSIDER DISPL CONSTR =', I5/1X, 'IDV=0 WILL NO
     2T CONSIDER FREQ. CONSTR =', I5/1X, 'IPD.EQ.0 WILL NOT PRINT DISPL
     3MATRIX AT EACH CYCLE=',I5/1X,'IPS=0 WILL NOT PRINT FORCE AND DISPL
     4 MATRIX AT EACH CYCLE=',I5,'          '/1X, 'IFS=NO OF TIMES STRESS
     *RATIO   DESIGN IS REQUIRED =', I5/1X, 'ITE = NUMBER  OF  ELEMENT
     6TYPE =',I5)
   27 FORMAT(//1X,'ILIM=LIMIT ON DESIGN CYCLES =', I5/1X, 'IPM=SUBSP MET
     1HOD ITRN LIMIT =', I5/1X, 'ITRS=NO OF TIMES STEP SIZE REDUCED =',
     2 I5/1X, 'LNSV=NO OF TIMES VARIATION IN COST FUN. REMAIN WITHIN SPE
     3CIFIED LIMITS =',I5)
   28 FORMAT(//1X,                         'DF IS REQ CHANGE
     1IN COST FUN =',E15.5/1X,   'RIT IS REQ CHANGE IN COST FUN WHEN ALL
     2CONSTRS ARE SATISFIED AND ILIM.GT.1 =',E15.5/1X,  'RIN IS REQ CHAN
     3GE IN COST FUN WHEN ALL CONSTR SATISFIED INITIALLY =',E15.5/1X,
     4'RL=SPECIFIED VARIATION IN COST FUN FOR REDUCING STEP SIZE =',E15.
     55/1X, 'EP IS EPCILON FOR CONSTRAINT CHECKS =',E15.5)
   29 FORMAT(//1X,'ERROR CRITERION-'/1X,'ERR1  EC FOR CONVERGENCE OF EVE
     1C. =',E15.5/1X,'ERR2  EC FOR TOLERANCE IN DELTA B1 NORM AT OPT. ='
     2,E15.5/1X,'ERR3  EC FOR TOLERANCE IN CONSTRS. AT OPT. =',E15.5/1X,
     3'ERR4  EC FOR TOLERANCE IN COST FUNCTION AT OPT. =',E15.5/1X,'ERR5
     4  EC FOR CHECKING ZERO ELEMENTS IN GAUSS. ELIMN. =',E15.5)
   30 FORMAT('0','    ***  DATA FOR INDIVIDUAL SUBSTRUCTURES K=',I2)
   31 FORMAT(/' ',2X,'*** SKIP DATA 28 THRU 31 AS NDAM=0')
   32 FORMAT(/' ', I3,'  (REDUC(I8),I8=LS,LE)')
   33 FORMAT(/'0',2X,' INVERSE OF WEIGHTING MATRIX. NORMALIZED WITH MAX.
     * ELEMENT.')
   34   FORMAT(//'   ***  DAMAGED CONDITION NO. I=',I2,'   ***')
```

```
35 FORMAT(/'1',' STRESS AND DISPLACEMENT VIOLATIONS'/' SIZE MEM/NC K
  * III GR  LOC IV  LC  BUC      1.0-XL'/)
36 FORMAT(I4,I6,7I4,F12.5)
37 FORMAT(1X,I3,4X,'EIGENVALUE         =',E16.7/8X,'NATURAL FREQUENCY='
  *,E16.7/8X,'EIGENVECTOR'/(4(I5,E12.4)))
39 FORMAT(' ****FREQUENCY IS NOT VIOLATED ****')
40 FORMAT(' **** FINAL RESPONSES ****')
45 FORMAT( /1X, 'STEP SIZE =',E15.7,I5)
46 FORMAT(2I3,2X,E13.5,8I5/(20X,8I5))
47 FORMAT(//1X,'SUBSTRUCTURE NO. ',2I5)
48 FORMAT(/1X,'VALUE OF COST FUNTION =',E16.7,' TRUSS=',E16.7,' CST='
  *,E16.7,' SSP=',E16.7//1X,                      'VALUES OF DESIGN
  1VARIABLES'/ 1X,'GR.NO.',4X,'AREA',11X,'MEMBER NUMBERS')
49 FORMAT('1', 'ITERATION NO =',4I4)
50 FORMAT(//1X,'COST FUNCTION HISTORY'/(4(I5,E12.4)))
51 FORMAT('1'////30X,'** INPUT DATA ERROR **')
52 FORMAT(/1X,I5,'  FULLY STRESSED DESIGN DESIRED INITIALLY, NO. OF
  1TIMES =',I4)
54 FORMAT( I4,2X,'BL VIOLATED, DV=',I3,2E15.5)
55 FORMAT( I4,2X,'BU VIOLATED, DV=',I3,2E15.5)
56 FORMAT(/1X,'TOTAL NO OF CONSTRAINTS VIOLATED =',I3)
58 FORMAT( /1X,'NO VIOLATION AT THIS ITERATION')
59 FORMAT( /1X,'NO CONSTRAINT VIOLATED INITIALLY NO OF TIMES=',I5)
60 FORMAT(' ','**** SKIP DATA 17 THRU 20 AS NCI=0 FOR K=',I2)
61 FORMAT(///'  IDC=',I2,', K=',I2,', III=',I2,', ITY(III)=',I2,' SKI
  *P DATA 21 THRU 23 IF ITY(III)=0.')
62 FORMAT(/' ', I3,'   NUNIT,NN,NSU,NDAM,NLC,NV,NCC,BNC,NBW,NPH,NSD,IS
  *PSP')
63   FORMAT(/' ',I3,'  IFS,IDV,IFR,IBUK,IDIS,IBDIS,IPS,IPD,IPC')
64 FORMAT(/' ', I3,'  ILIM,ITRS,LNSV,      LCON,ICONT,(ITY(I),I=1,ITE)
  1,IWMM')
65 FORMAT(/' ', I3,'       DF,RIT,RIN,RL,EP,STP1,STP2')
66 FORMAT(/' ', I3,'  (FACC(I),I=1,ITE),RF,CONL')
67   FORMAT(/' ', I3,'   RRF(I),RDLIM(I),RSL(I),RSU(I),RLOAD(I)')
68 FORMAT(/' ', I3,'  ERR1,ERR2,ERR3,ERR4,ERR5,FACTOR')
69 FORMAT(/' ', I3,'  (DLIB(I),I=1,BNC)')
70 FORMAT(/' ',3I3,'  NLJ - (NJL(I),I=1,NLJ) - J,(PB(N,L),N=1,NN) ---
  * FOR ALL NLC.')
71 FORMAT(11X,'ZZ(I,1) IS',15X,'ZZ(I,2) IS'/6X,'TRANLAMBDA*DELTAB1',5
  1X,'TRANLAMBDA*DELTAB2'/14X,'=0',19X,'=DELPHI')
72 FORMAT(I4,2X,E16.7,7X,E16.7)
73 FORMAT( /1X,'(CHANGE IN COST FUNCTION =',E15.5//1X,'DB(I)*DB(I)
  1=',E15.5)
74 FORMAT(/1X,'T(2) IS TRANDELTAB1*DELTAB2=',E16.7//1X,'T(3) IS TRANL
  1J*DELTAB1=',E16.7)
75 FORMAT(/1X,'DELTA B1 NORM HISTORY'/(4(I5,E12.4)))
76 FORMAT(/1X,'DELTA B2 NORM HISTORY'/(4(I5,E12.4)))
77 FORMAT( /1X,'NUMBER OF TIMES COST VARIATION'/1X,'REMAINS WITHIN SP
  1ECIFIED LIMITS=',I5)
78 FORMAT(//1X,'NO OF TIMES STEP SIZE REDUCED=',I3/1X,'NEW STEP SIZE
  1=',E15.7/1X,'REQ CHANGE IN COST FUNCTION DF=',E15.7)
79 FORMAT( '1','CONVERGENCE CRITERIA HAS BEEN SATISFIED')
81 FORMAT(3X,'NV    DELTAB1          DELTAB2          DELTAB')
82 FORMAT(I5,3E14.5)
83 FORMAT(/' ', I3,'  LINK')
84 FORMAT(/' ', I3,'  LINLG(I,1),LINLG(I,2) - SKIP IF LINK=0')
85 FORMAT(/' ', I3,'  NJ(K),NBJ(K),NCB(K),NIC(K),NBW1(K),NBW2(K),NBW3
  *(K)')
86 FORMAT(/' ', I3,'  (NZ(I,K),I=1,NB)')
87 FORMAT(/' ', I3,'  JN,X(J,K),Y(J,K),Z(J,K),(ND(I),I=1,NN) - FOR AL
  *L NJK')
```

```
   88 FORMAT(/' ', I3,'  (DLIM(I,K),I=1,NCI)')
   89 FORMAT(/' ',3I3,'  NLJ - (NJL(I),I=1,NLJ) - J,(PI(N,L,K),N=1,NN) -
     *-- FOR ALL NLC. SKIP 19 AND 20 IF NLJ=0')
   90 FORMAT(/' ', I3,'  NM(KK),NG(KK),NW(KK),MEB(KK),MEF(KK)')
   91 FORMAT(/' ', I3,'  (B(I,KK),I=1,NGK)')
   92 FORMAT(/' ', I3,'   (IGRT(I,KK),I=1,NGK)')
   93 FORMAT(/' ', I3,'  J,L,(MN(N+M,KK),M=1,L)')
   94 FORMAT(/' ', I3,'   EVEC FOR')
   95 FORMAT(/' ', I3,'  CL(I,KK),BU(I,KK),ALP(I,KK),SL(I,KK),SU(I,KK),R
     *J(I),XNUU(I,KK),E(I,KK)')
   96 FORMAT(/' ', I3,'  M8,JP,JQ,JR')
   97   FORMAT(/' ', I3,'  K=',I2,', KIIDAM(K,I)=',I2)
   98 FORMAT(/' ', I3,'  N - SKIP DATA 30 AND 31 IF N=0')
   99 FORMAT(/' ', I3,'  (NDM(I8),I8=LS,LE)')
C
C.... A-DATA COMMON TO ALL SUBSTRUCTURES.
C
      PIS=(3.1415927)**2
      CCC=1.0
      IPM=0
      ICHEK=0
      GG=386400.0
      WRITE(6,24)
      NUMBER=1
      WRITE(6,62) NUMBER
      NUMBER=NUMBER+1
      ITE=3
10001 READ(5,10) NUNIT,NN,NSU,NDAM,NLC,NV,NCC,BNC,NBW,NPH,NSD,ISPSP
      WRITE(6,10)  NUNIT,NN,NSU,NDAM,NLC,NV,NCC,BNC,NBW,NPH,NSD,ISPSP
      IF(NUNIT.EQ.1) GG=1.0
      SN=2*NN
      NN1=NN-1
      WRITE(6,63) NUMBER
      NUMBER=NUMBER+1
10002 READ(5,10,ERR=777) IFS,IDV,IFR,IBUK,IDIS,IBDIS,IPS,IPD,IPC,JUSTW,
     2IAUTO
      WRITE(6,10)        IFS,IDV,IFR,IBUK,IDIS,IBDIS,IPS,IPD,IPC,JUSTW,
     2IAUTO
      WRITE(6,64) NUMBER
      NUMBER=NUMBER+1
10003 READ(5,10,ERR=777) ILIM,ITRS,LNSV,      LCON,ICONT,
     *(ITY(I),I=1,ITE),IWMM
      WRITE(6,10)        ILIM,ITRS,LNSV,      LCON,ICONT,
     *(ITY(I),I=1,ITE),IWMM
      WRITE(6,65) NUMBER
      NUMBER=NUMBER+1
10004 READ(5,11,ERR=777)    DF,RIT,RIN,RL,EP,STP1,STP2
      WRITE(6,11)     DF,RIT,RIN,RL,EP,STP1,STP2
      WRITE(6,66) NUMBER
      NUMBER=NUMBER+1
C
C     FACC(I) WILL BE USED LATER TO GENERATE WEIGHTING MATRIX W
C
10005 READ(5,11)   (FACC(I),I=1,ITE),RF,CONL
      WRITE(6,11) (FACC(I),I=1,ITE),RF,CONL
      WRITE(6,68) NUMBER
      NUMBER=NUMBER+1
10006 READ(5,8,ERR=777) ERR1,ERR2,ERR3,ERR4,ERR5
      WRITE(6,8)        ERR1,ERR2,ERR3,ERR4,ERR5
      WRITE(6,69) NUMBER
      NUMBER=NUMBER+1
```

```
10007 READ(5, 5,ERR=777) (DLIB(I),I=1,BNC)
      WRITE(6,11)         (DLIB(I),I=1,BNC)
      DO 105 I=1,BNC
  105 DLIB(I)=1.DO/DLIB(I)
      L=9
      I=10
      WRITE(6,70) NUMBER,L,I
      NUMBER=NUMBER+3
C
C     BOUNDARY LOAD,HOWEVER SHOULD CONSIDER JUST 'ACTIVE'! DOF ONLY
C
      DO111 L=1,NLC
      DO109 I=1,BNC
  109 PB(I,L)=0.DO
10008 READ(5,10,ERR=777) NLJ
      WRITE(6,10)        NLJ
      IF(NLJ.EQ.0) GO TO 111
10009 READ(5,10,ERR=777)       (NJL(I),I=1,NLJ)
      WRITE(6,10)              (NJL(I),I=1,NLJ)
      DO110 I=1,NLJ
      LE=NN*NJL(I)
      LS=LE-NN1
10010 READ(5,9,ERR=777) J,(PB(N,L),N=LS,LE)
  110 WRITE(6,9)        J,(PB(N,L),N=LS,LE)
  111 CONTINUE
      WRITE(6,83) NUMBER
      NUMBER=NUMBER+1
10011 READ(5,10,ERR=777)LINK
      WRITE(6,10) LINK
      WRITE(6,84) NUMBER
      NUMBER=NUMBER+1
      IF(LINK.EQ.0) GO TO 131
      DO 130 I=1,LINK
10012 READ(5,10,ERR=777) LINLG(I,1),LINLG(I,2)
  130 WRITE(6,10)        LINLG(I,1),LINLG(I,2)
  131 LIN=0
      DO 132 I=1,ITE
  132 NVV(I)=0
C
C.... B-DATA FOR INDIVIDUAL SUBSTRUCTURES
C
      MM=0
      LQ=0
      KK=0
      I8=0
      I9=0
      J8=0
      M8=0
C
C     BEGIN FOR BIG LOOP 7777
C
      DO 7777 K=1,NSU
      KIIDAM(K,1)=1
      WRITE(6,30) K
      NUMBER=13
      WRITE(6,85) NUMBER
      NUMBER=NUMBER+1
10013 READ(5,10) NJ(K),NBJ(K),NCB(K),NIC(K),NBW1(K),NBW2(K),NBW3(K)
      WRITE(6,10) NJ(K),NBJ(K),NCB(K),NIC(K),NBW1(K),NBW2(K),NBW3(K)
C
      CALL VARI(K)
```

```
      WRITE(6,86) NUMBER
      NUMBER=NUMBER+1
C
C     TO CONVERT BOUNDARY NODES,FROM LOCAL TO OVER ALL NUMBERING SYSTEM
C
10014 READ(5,10,ERR=777) (NZ(I,K),I=1,NB)
      WRITE(6,10)        (NZ(I,K),I=1,NB)
      WRITE(6,87) NUMBER
      NUMBER=NUMBER+1
      DO 140 J=1,NJK
      LE=NN*J
      LS=LE-NN1
10015 READ(5,9,ERR=777) JN,X(J,K),Y(J,K),Z(J,K),(ND(I),I=LS,LE)
  140 WRITE(6,9)        JN,X(J,K),Y(J,K),Z(J,K),(ND(I),I=LS,LE)
      IF(NCI.EQ.0) GO TO 156
      WRITE(6,88) NUMBER
      NUMBER=NUMBER+1
10016 READ(5,11,ERR=777) (DLIM(I,K),I=1,NCI)
      WRITE(6,11)        (DLIM(I,K),I=1,NCI)
      DO 141 I=1,NCI
  141 DLIM(I,K)=1.D0/DLIM(I,K)
      L=18
      I=19
      WRITE(6,89) NUMBER,L,I
      NUMBER=NUMBER+3
      DO 155 L=1,NLC
      DO150 I=1,NCI
  150 PI(I,L,K)=0.D0
10017 READ(5,10,ERR=777) NLJ
      WRITE(6,10)        NLJ
      IF(NLJ.EQ.0) GO TO 155
C
C     INTERIOR LOAD, I DOF WILL BE SUBTRACTED BY NCB(K) TO SAVE MEMORY
C
10018 READ(5,10,ERR=777)      (NJL(I),I=1,NLJ)
      WRITE(6,10)             (NJL(I),I=1,NLJ)
      DO 154 I=1,NLJ
      LE=NN*NJL(I)-N1
      LS=LE-NN1
10019 READ(5,9,ERR=777) J,(PI(N,L,K),N=LS,LE)
  154 WRITE(6,9)        J,(PI(N,L,K),N=LS,LE)
  155 CONTINUE
      GO TO 157
  156 WRITE(6,60) K
C
C     TO GENERATE B DOFIN OVER ALL SYSTEM,
C     HOWEVER SHOULD CONSIDER JUST 'ACTIVE'B DOF ONLY
C
  157 DO 160 I=1,NB
      L=NZ(I,K)
      L1=NN*(L-1)
      I1=NN*(I-1)
      DO 160 J=1,NN
      L1=L1+1
      I1=I1+1
      NZC(I1,K)=L1
  160 CONTINUE
C
C     CUMULATIVE RESTRAINT LIST
C
      NJJ=NJK*NN
```

```
      I=0
      DO161 J=1,NJJ
      I=ND(J)+I
  161 ND(J)=ND(J)*I
      IF(I.EQ.NC) GO TO 162
      WRITE(6,12)
      GO TO 222
C
C.... DATA FOR INDIVIDUAL FINITE ELEMENTS.
C
  162 DO 7777 III=1,ITE
      NUMBER=20
      IDC=0
      WRITE(6,61) IDC,K,III,ITY(III)
      IF(ITY(III).EQ.0) GO TO 7777
      KK=KK+1
      WRITE(6,90) NUMBER
      NUMBER=NUMBER+1
10020 READ(5,10,ERR=777) NM(KK),NG(KK),NW(KK),MEB(KK),MEF(KK)
      WRITE(6,10)         NM(KK),NG(KK),NW(KK),MEB(KK),MEF(KK)
      NGK=NG(KK)
      MA=0
      N=0
      WRITE(6,93) NUMBER
      NUMBER=NUMBER+1
      DO180 I=1,NGK
10021 READ(5,10,ERR=777) J,L,(MN(N+M,KK),M=1,L)
      WRITE(6,10)         J,L,(MN(N+M,KK),M=1,L)
      DO 179 LL=1,L
      MA=MA+1
      M=MN(MA,KK)
  179 IGRE(M,III)=I
      N=N+L
  180 NOM(I,KK)=L
      M6=MEB(KK)
      M7=MEF(KK)
      WRITE(6,95) NUMBER
      NUMBER=NUMBER+1
      DO 181 I=1,NGK
10022 READ(5,11,ERR=777)           BL(I,KK),BU(I,KK),ALP(I,KK),SL(I,KK),
     1SU(I,KK),RO(I),    XNUU(I,KK),E(I,KK)
      WRITE(6,11)                  BL(I,KK),BU(I,KK),ALP(I,KK),SL(I,KK),
     1SU(I,KK),RO(I),    XNUU(I,KK),E(I,KK)
      SL(I,KK)=1.0D0/SL(I,KK)
  181 SU(I,KK)=-1.0D0/SU(I,KK)
      WRITE(6,96) NUMBER
      NUMBER=NUMBER+1
C
10023 CALL ELESTF(M5,III,I8,K,KK,M6,M7,I9,ISPSP,NN,J8,M8,IDV,GG)
      DO 193 I=M6,M7
      IGR=IGRE(I,III)
  193 RR(I,III)=RO(IGR)*ELL(I,III)
      IF(IBUK.EQ.0.OR.III.GT.1) GO TO 7777
      DO 194 I=1,NGK
      BUC=ALP(I,KK)*E(I,KK)*PIS
  194 E(I,KK)=1.D0/BUC
 7777 CONTINUE                                                          ****
C
C     END OF BIG LOOP 7777
C
C
C
```

```
C       KIIDAM(K,IDC).EQ.0   - NOT DAMAGED.
C       KIIDAM(K,IDC).NE.0   -     DAMAGED.
C
        IPDAM=NDAM+1
        RRF(1)=1.0
        RDLIM(1)=1.0
        RSL(1)=1.0
        RSU(1)=1.0
        RLOAD(1)=1.0
        NEGV(1)=0
        NDOF(1)=NCC
        LS=1
        LE=0
        IDC=0
        IF(NDAM.EQ.0) GO TO 201
C
C       INPUT DAMAGED DISCRIPTION
C
        DO 200 IDC=1,NDAM
        WRITE(6,34) IDC
        I=IDC+1
        NEGV(I)=0
        NUMBER=24
        WRITE(6,67) NUMBER
10024   READ(5,11,ERR=777) RRF(I),RDLIM(I),RSL(I),RSU(I),RLOAD(I)
        WRITE(6,11)        RRF(I),RDLIM(I),RSL(I),RSU(I),RLOAD(I)
        KK=0
        DO 200 K=1,NSU
        NUMBER=25
10025   READ(5,10) KIIDAM(K,I)
        WRITE(6,97)NUMBER,K,KIIDAM(K,I)
        DO 200 III=1,ITE
        NUMBER=26
        IF(ITY(III).EQ.0) GO TO 200
        KK=KK+1
        WRITE(6,98) NUMBER
        NUMBER=NUMBER+1
10026   READ(5,10,ERR=777) N
        WRITE(6,10)        N
        NBDAM(KK,IDC)=N
        NEGV(I)=NEGV(I)+N
        IF(N.EQ.0) GO TO 200
        LE=LE+N
        WRITE(6,99) NUMBER
        NUMBER=NUMBER+1
10027   READ(5,10,ERR=777) (NDM(I8),I8=LS,LE)
        WRITE(6,10)        (NDM(I8),I8=LS,LE)
        WRITE(6,32) NUMBER
10028   READ(5,5,ERR=777) (REDUC(I8),I8=LS,LE)
        WRITE(6,5)        (REDUC(I8),I8=LS,LE)
        LS=LS+N
  200   CONTINUE
        GO TO 202
  201   WRITE(6,31)
 202    IF(IDV.EQ.0  .AND.  JUSTW.EQ.0)GO TO 204
        IF(IAUTO.EQ.0)GO TO 864
C
C       AUTOMATIC GENERATION OF INPUT EIGEN VECTOR(FOR SUB. SUBSP OPTION)
C
        KU=0
        DO 865 ITWO=1,2
```

```
      DO 866 J=1,NCC
      KU=KU+1
      IF(KU.EQ.ITWO)XEIG(J,ITWO)=1.D0
      IF(KU.NE.ITWO)XEIG(J,ITWO)=0.D0
 866  CONTINUE
      KU=0
 865  CONTINUE
      GO TO 204
C
C     OR USER HAS TO SUPPLY INPUT EV(FOR SUB. SUBSP OPTION)
C
 864  NUMBER=29029
      WRITE(6,94)NUMBER
      DO 803 ITWO=1,2
29029 READ(5,8,ERR=777)(XEIG(J,ITWO),J=1,NCC)
 803  WRITE(6,8)        (XEIG(J,ITWO),J=1,NCC)
 204  KK=0
      DO 210 K=1,NSU
      DO 210 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 210
      KK=KK+1
      NGK=NG(KK)
      NUMBER=30
      WRITE(6,91) NUMBER
10030 READ(5, 4,ERR=777)  (B(I,KK),I=1,NGK)
      NUMBER=NUMBER+1
      WRITE(6, 4)         (B(I,KK),I=1,NGK)
      WRITE(6,92) NUMBER
10031 READ(5,10,ERR=777)  (IGRT(I,KK),I=1,NGK)
      WRITE(6,10)         (IGRT(I,KK),I=1,NGK)
C
C     2 SMALL LOOPS 207 & 208 TO GENERATE DV = FOR COMPLETE STRUCTURE
C     STORED IN IGRT(-,-),ALSO IGRE(-,-)CONTAINS GROUPE =
C     IN SUBSTRUCTURE K,ELEMENT TYPE III
C
      DO 207 I=1,NGK
      IF(IGRT( I,KK)) 206,207,205
 205  NVV(III)=NVV(III)+1
      IGRT( I,KK)=NVV(III)
      GO TO 207
 206  LIN=LIN+1
      LLL=LINLG(LIN,1)
      NGR=LINLG(LIN,2)
      IGRT( I,KK)=IGRT(NGR,LLL)
 207  CONTINUE
 210  CONTINUE
      NA(1)=NVV(1)
      NA(2)=NVV(1)+NVV(2)
      KK=0
      DO 208 K=1,NSU
      DO 208 III=1,ITE
      IF(ITY(III).EQ.0)GO TO 208
      KK=KK+1
      IF(III.EQ.1)GO TO 208
      NGK=NG(KK)
      DO 209 I=1,NGK
      IF(IGRT(I,KK).EQ.0)GO TO 209
      IGRT(I,KK)=IGRT(I,KK)+NA(III-1)
 209  CONTINUE
 208  CONTINUE
```

```
C       BEGIN TO GENERATE WEIGHTING MATRIX
C
        DO 211 I=1,NV
  211   WM(I)=0.0D0
        KK=0
        DO 220 K=1,NSU
        DO 220 III=1,ITE
        IF(ITY(III).EQ.0) GO TO 220
        KK=KK+1
        M6=MEB(KK)
        M7=MEF(KK)
        DO 219 I=M6,M7
C       IF(III.EQ.1) X(I,1)=H(I)
        MV=IGRT(IGRE(I,III),KK)
        IF(MV.EQ.0) GO TO 219
        OO(MV)=FACC(III)
        WM(MV)=WM(MV)+RR(I,III)
  219   CONTINUE
  220   CONTINUE
        DO 221 I=1,NV
        AXL=OO(I)
         OO(I)=WM(I)
  221   WM(I)=WM(I)*AXL
        XX=WM(1)
        DO 230 I=2,NV
        IF(XX.GE.WM(I)) GO TO 230
        XX=WM(I)
  230   CONTINUE
        DO 231 I=1,NV
  231   WM(I)=XX/WM(I)
C
C.... WM(I) STORES INVERSE OF WEIGHTING MATRIX. NORMALIZED WITH MAX ELE.
C
        WRITE(6,33)
        WRITE(6,10) (NVV(I),I=1,ITE)
        WRITE(6,8) (WM(I),I=1,NV)
        IF(IWMM.EQ.0) GO TO 233
        DO 232 I=1,NV
  232   WM(I)=1.0D0
  233   SUML=0.0
        WRITE(6,8) (WM(I),I=1,NV)
        DO 234 I=1,NV
  234   SUML=SUML+OO(I)*WM(I)*OO(I)
        WRITE(6,24)
        WRITE(6,25) SN,NSU,BNC, NBW,NLC,NPH,NSD
        WRITE(6,26)  IBUK, IDIS,IDV,IPD,IPS,IFS,ITE
        WRITE(6,27) ILIM,IPM,ITRS,LNSV
        WRITE(6,28)DF,RIT,RIN,RL,EP
        WRITE(6,29) ERR1,ERR2,ERR3,ERR4,ERR5
C
C       INITIALIZE COUNTERS
C
        NTL=0
        ITRN=0
        ITR=0
        ICV=0
        NSV=0
  998   CONTINUE
C
C       COST FUNCTION & STEP SIZE
C
```

```
      DO 240 I=1,ITE
  240 XCOST(I)=0.0
      K=0
      DO 242 KK=1,NSU
      DO 242 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 242
      K=K+1
      M6=MEB(K)
      M7=MEF(K)
      DO 241 J=M6,M7
      I=IGRE(J,III)
  241 XCOST(III)=XCOST(III)+B(I,K)*RR(J,III)
  242 CONTINUE
      COST=0.0
      DO 245 I=1,ITE
  245 COST=COST+XCOST(I)*CCC
      STEP=(COST*DF)/(CCC*SUML)
      STEP=STEP*STP2
  999 CONTINUE
      NTL=NTL+1
      FB(NTL)=COST
      DBIN(NTL,1)=0.0D0
      DBIN(NTL,2)=0.0D0

      PRINTING COST FN. HISTORY

      WRITE(6,49) NTL
      WRITE(6,45) STEP
      WRITE(6,48)  COST,(XCOST(I),I=1,3)
      IF(NTL.GT.ILIM) GO TO 22220

      PRINTING CURRENT AREAS

 7007 K=0
      DO 246 I=1,NCC
  246 NDISP(I)=0
      DO 248 KK=1,NSU
      DO 248 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 248
      K=K+1
      WRITE(6,47) KK,III
      MA=0
      N=NG(K)
      DO 247 I=1,N
      NGV(I,K)=0
      J=NOM(I,K)
      WRITE(6,46) I,IGRT(I,K),B(I,K),(MN(MA+L,K),L=1,J)
  247 MA=MA+J
  248 CONTINUE
      DO 7008 NFVIO=1,IPGAM
      TEI(NFVIO)=0.D0
 7008 TE(NFVIO)=0.D0
      SIZE=0
      LX=0
      INCR=0
      NORES=0
      I1X8=0
      I3X8=0
      I4X8=0
      DO 256 I=1,NV
  256 SS(I)=0.0
```

```
C
C     BEGIN OF BIG LOOP 77788  FREQ ANAL & CHECK IF FREQ IS(ARE) VIOLATED
C
      DO 77788 IDC=1,IPDAM
      LDC=IDC-1
      WRITE(6,34)LDC
      DO 259 I=1,BNC
      DO 258 J=1,NBW
  258   C(I,J)=0.DO
      DO 259 L=1,NLC
  259 ZB(I,L)=PB(I,L)*RLOAD(IDC)
      IF(IDV.EQ.0 .AND. JUSTW.EQ.0)GO TO 2550
      XRF=RF*RRF(IDC)
      XRFF=(6.2831854*XRF)**2
      IO12=0
      CALL STIFFM(N,K,IDC,IIX8,&883,IO12)
      GO TO 882
  883 WRITE(6,19)N,K,IDC
      GO TO 222
  882 DO 832 I=1,BNC
      DO 832 J=1,NBW
  832 D(I,J)=C(I,J)
      N=0
      K=0
      CALL DECUPP(N,NBW,BNC,&884)
      GO TO 885
  884 WRITE(6,19)N,K
      GO TO 222
  885 CONTINUE
      DO 850 I=1,BNC
      DO 850 J=1,NBW
  850 C(I,J)=D(I,J)
      CALL SUBSP(NCC,NBW,LCON,ERR1,IDC,I3X8)
      FREQ=WS(1)
      XL=DSQRT(FREQ)/6.2831853
      WRITE(6,37)NCC,FREQ,XL,(I,XEIG(I,1),I=1,NCC)
      IF(IDV.EQ.0 .AND. JUSTW.NE.0)GO TO 2550
      YYM=1.0-(FREQ/XRFF)+EP
      IF(ICHEK.EQ.1)GO TO 2550
      IF(YYM.LT.0.0)GO TO 254
      LX=LX+1
      TEI(LX)=(FREQ-XRFF)/XRFF
      TE(LX)=DABS(XRF-XL)/XRF
      KK=0
      DO 251 K=1,NSU
      DO 251 III=1,3
      IF(ITY(III).EQ.0) GO TO 251
      KK=KK+1
      M6=MEB(KK)
      M7=MEF(KK)
      DO 252 I4=M6,M7
  252 BE(I4,III)=BR(I4,III)
  251 CONTINUE
      IF(ICHEK.EQ.1)GO TO 2550
C
C     TO FIND TRIPLE PRODUCT Y*M*Y,USE LATER IN SUB. DEFREQ
C
      CALL MEVEC(NM,NCC,IDC,I4X8,1)
      FDEN=0.DO
      DO 879 I=1,NCC
  879 FDEN=FDEN+XEIG(I,1)*YXEIG(I,1)
```

```
      CALL DEFREQ(FREQ,XRFF,NN,FDEN,NCC)
      DO 253 I=1,NV
253   ETC(I+INCR)=-H(I)
      INCR=INCR+NV
      IF((IFR.GT.0).AND.(ITRN.EQ.0)) GO TO 332
      GO TO 2551
 254  NORES=1
      WRITE(6,39)
2550  I4X8=I4X8+NEGV(IDC)
551   CONTINUE

      BEGIN OF BIG LOOP---88    TO FIND DISPL,EL. FORCE
      CONSTRAINT CHECK ON STRESS,DISPL,CONSTRUCT CAP LAMDA MATRIX
      =DERIVATIVE OF VIOLATED CONSTRAINTS


      IF(IDV.EQ.0 .AND. JUSTW.EQ.0)GO TO 820
      GO TO 821
820   CALL STIFFM(N,K,IDC,IIX8,&260,0)
      GO TO 270
 260  WRITE(6,19) N,K,IDC
      GO TO 222
 270  DO 302 I=1,BNC
      DO 302 J=1,NBW
 302  D(I,J)=  C(I,J)
      N=0
      K=0

      CALL DECUPP(N,NBW,BNC,&303)
      GO TO 304
 303  WRITE(6,19) N,K
      GO TO 222
 304  CONTINUE

      GO TO 837
821   CONTINUE
      DO 903 I=1,BNC
      DO 903 J=1,NBW
903   D(I,J)=C(I,J)
837   CALL  ZBZIEF(IDC,ISPSP,IPS,IPD)
      IF(ICHEK.EQ.1)GO TO 77788
      DO 310 I=1,NLC
 310  NO(I)=0

      CALL CONST(IDC,IBUK,IDIS,IBDIS,NSD,EP,MV,IBU,IV,IPC,NTL,IFS,ISPSP,
     1NDAM)
      IF( NTL.GT.IFS) GO TO 311
      WRITE(6,52)     NTL,IFS
      ITRN=ITRN+1
      GO TO 998
311   IF(IV.EQ.0)GO TO 77788
      DO 331 I=1,BNC
      DO 330 J=1,NBW
 330  D(I,J)=C(I,J)
      DO 331 J=1,IV
 331  DS(I,J)=A2(I,J)

      CALL SOLDUP(IV,NBW,BNC)
      WRITE(6,311) ((DS(I,J),I=1,BNC),J=1,IV)

      LXX=1
```

```
      CALL GENC(NSD,NV,LXX,IBU,IBUK,IV,IDC)
      IF(SIZE.GT.(NSD-NDAM-1))GO TO 77789
77788 CONTINUE
C
C     END OF BIG LOOP 77788
C
77789 CONTINUE
      IF(ICHEK.EQ.1) GO TO 222
      IF(LX.EQ.0 .AND. SIZE.EQ.0)GO TO 332
      IF(SIZE.EQ.0 .AND. LX.NE.0)GO TO 33318
      WRITE(6,35)
      DO 318 I=1,SIZE
      WRITE(6,36)I,(INF(I,J),J=1,8),DLPH(I)
 318  CONTINUE
33317 DO 319 I=1,SIZE
      DO 319 J=1,NV
  319 DPB(J,I)=DPX(J,I)
      IF(LX.EQ.0 .AND. SIZE.NE.0)GO TO 332
C
C     ADD COLUMNS OF FREQ. VIOLATIONS IN CAP LAMDA MATRIX
C     ALSO ADD AMOUNT OF FREQ VIOL IN DELPHI,THEN UPDATED SIZE
C
33318 INCR=-NV
      DO 320 I=1,LX
      DLPH(SIZE+I)=TEI(I)
      DLP(SIZE+I)=TE(I)
      INCR=INCR+NV
      DO 320 J=1,NV
 320  DPB(J,SIZE+I)=ETC(J+INCR)
      SIZE=SIZE+LX
C
C.... CHECK FOR DESIGN VARIABLE CONSTRAINTS
C
  332 IJ=SIZE
      NDC=0
      K=0
      DO 334 I=1,NV
      Z(I,1)=0.0
  334 VV(I)=0.0
      DO343 KK=1,NSU
      DO 343 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 343
      WRITE(6,47) KK,III
      K=K+1
      LL=0
      NGK=NG(K)
      DO 342 I=1,NGK
      L=NOM(I,K)
      MA=IGRT(I,K)
      IF(MA.EQ.0) GO TO 341
      IF(SS(MA).EQ.1.0.OR.VV(MA).EQ.1.0) GO TO 339
      VV(MA)=1.0
      IF(BL(I,K).LE.0.) GO TO 335
      YYM=1.0
      ZN=BL(I,K)
      XL=B(I,K)/ZN
      GO TO 336
  335 YYM=0.
      ZN=1.0
      XL=B(I,K)
  336 CONTINUE
```

```
      IF((YYM+EP).LT.XL) GO TO 337
      SIZE=SIZE+1
      IF(SIZE.LE.NPH)GO TO 3380
      SIZE=SIZE-1
      GO TO 338
 3380 NDC=NDC+1
      DLPH(SIZE)=XL-YYM
      DLP(SIZE)=-DLPH(SIZE)
      WRITE(6,54) SIZE,MA,XL
      DZE(NDC)=-1.D0/ZN
      GO TO 338
  337 CONTINUE
      ZN=BU(I,K)
      XL=B(I,K)/ZN
      IF((XL+EP).LT.1.0) GO TO 339
      SIZE=SIZE+1
      IF(SIZE.LE.NPH)GO TO 3381
      SIZE=SIZE-1
      GO TO 338
 3381 NDC=NDC+1
      DLPH(SIZE)=1.D0-XL
      DLP(SIZE)=-DLPH(SIZE)
      WRITE(6,55) SIZE,MA,XL
      DZE(NDC)=1.D0/ZN
  338 H(NDC)=MA
  339 CONTINUE
      DO 340 J=1,L
      LL=LL+1
      M=MN(LL,K)
  340 Z(MA,1)=Z(MA,1)+RR(M,III)
      GO TO 342
  341 LL=LL+L
  342 CONTINUE
  343 CONTINUE
      IF(SIZE.EQ.0) GO TO 347
      WRITE(6,56) SIZE
C.... COMPUTE DELTA B VECTOR
C.... DPB IS CAP LAMBDA MATRIX (NV,NSD)
      IF(IJ.EQ.0) GO TO 345
      DO 344 J=1,NV
      RO(J)=DSQRT(WM(J))
      DO 344 I=1,IJ
  344 DPB(J,I)=DPB(J,I)*RO(J)
  345 CONTINUE
C
      CALL DELBE(IJ,NDC,NV,&347)
      IF(IJ.EQ.0) GO TO 351
      DO 346 J=1,NV
      XX=1.0D0/RO(J)
      DO 346 I=1,IJ
  346 DPB(J,I)=DPB(J,I)*XX
      GO TO 351
  347 CONTINUE
C...  NO VIOLATION OF CONSTRAINTS
      IF(ITRN.EQ.0) GO TO 348
      WRITE(6,58)
      XL=RIT
      DF=XL
      STEP=(COST*XL)/(CCC*SUML)
      STEP=STEP*STP2
      GO TO 349
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
  348 XL=RIN
C.... NO INITIAL VIOLATION
      ICV=ICV+1
      WRITE(6,59)    ICV
  349  YYM=(COST*XL)/(CCC*SUML)
      YYM=YYM*STP2
      WRITE(6,45) YYM
      DO 350 I=1,NV
      BE(I,1)=-Z(I,1)*WM(I)
      BE(I,2)=0.
  350 W(I)=YYM*BE(I,1)
  351 CONTINUE
C.... COMPUTATIONAL CHECKS.
      IF(IJ.EQ.0) GO TO 354
      WRITE(6,71)
      DO353 I=1,IJ
      ZZ(I,1)=0.0
      ZZ(I,2)=0.0
      DO352 J=1,NV
      ZZ(I,1)=ZZ(I,1)+DPP(J,I)*BE(J,1)
  352 ZZ(I,2)=ZZ(I,2)+DPP(J,I)*BE(J,2)
  353 WRITE(6,72)    I,ZZ(I,1),ZZ(I,2)
  354 DO 355 I=1,5
  355 T(I)=0.0D0
      DO356 I=1,NV
      FF=1.0D0/WM(I)
      T(1)=T(1)+BE(I,1)*BE(I,1)*FF
      T(2)=T(2)+BE(I,1)*BE(I,2)*FF
      T(4)=T(4)+BE(I,2)*BE(I,2)*FF
  356 T(3)=T(3)+Z(I,1)*  BE(I,1)
      DBIN(NTL,1)=DSQRT(T(1))
      DBIN(NTL,2)=DSQRT(T(4))
C.... COMPUTE NEW B
      IF(SIZE.EQ.0)GO TO 358
      DO 357 I=1,NV
 357  W(I)=STEP*BE(I,1)+BE(I,2)
  358 SUM=0.0
      DO 359 I=1,ITE
  359 XCOST(I)=0.0
      DO 360 I=1,NV
  360 VV(I)=0.0
      LIN=0
      K=0
      DO 364 KK=1,NSU
      DO 364 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 364
      K=K+1
      V=0
      NGK=NG(K)
      DO 363 I=1,NGK
      L=IGRT(I,K)
      IF(L.EQ.0) GO TO 362
      IF(VV(L).EQ.1.0) GO TO 361
      SUM=SUM+W(L)*W(L)
      B(I,K)=B(I,K)+W(L)
      IF(B(I,K).LT.BL(I,K)) B(I,K)=BL(I,K)
      IF(B(I,K).GT.BU(I,K))B(I,K)=BU(I,K)
      VV(L)=1.0
      GO TO 362
  361 LIN=LIN+1
      LLL=LINLG(LIN,1)
```

```
      NGR=LINLG(LIN,2)
      B(I,K)=B(NGR,LLL)
  362 NJJ=NOM(I,K)
      DO 363 J=1,NJJ
      N=N+1
      M=MN(N,K)
  363 XCOST(III)=XCOST(III)+B(I,K)*RR(M,III)
  364 CONTINUE
      VALUE=0.0D0
      DO 365 I=1,ITE
  365 VALUE=VALUE+XCOST(I)*CCC
      XL=(COST-VALUE)/COST
      WRITE(6,73)    XL,SUM
      XL=DABS(XL)
      WRITE(6,81)
      DO 366 I=1,NV
  366 WRITE(6,82)  I,BE(I,1),BE(I,2),W(I)
      WRITE(6,74)    T(2),T(3)
      WRITE(6,75)    (I,DBIN(I,1),I=1,NTL)
      WRITE(6,76)    (I,DBIN(I,2),I=1,NTL)
      WRITE(6,50) (I,FB(I),I=1,NTL)
      IF(ITR.EQ.ITRS) GO TO 369
      IF(XL.GT.RL) GO TO 368
      NSV=NSV+1
      WRITE(6,77)    NSV
      IF(NSV-INSV) 369,367,367
  367 ITR=ITR+1
      DF=STP1*DF
      RIT=STP1*RIT
      RL=0.5*RL
      STEP=(COST*DF)/(CCC*SUML)
      STEP=STEP*STP2
      WRITE(6,78)  ITR,STEP,DF
  368 NSV=0
  369 CONTINUE
      IF((SIZE.EQ.0).AND.(ITRN.EQ.0)) GO TO 998
      IF(SIZE.EQ.0) GO TO 371
      DO 370 I=1,SIZE
      IF(DLP(I).GT.ERR3) GO TO 371
  370 CONTINUE
      IF(DBIN(NTL,1).LT.ERR2) GO TO 372
  371 COST=VALUE
      ITRN=ITRN+1
      GO TO 999
  372 WRITE(6,79)
      WRITE(6,48) VALUE,(XCOST(I),I=1,3)
      ICHEK=1
      WRITE(6,40)
      GO TO 7007
22220 K=0
      IF(IDV.EQ.0 .AND. JUSTW.EQ.0)GO TO 379
      DO 378 I=1,2
  378 WRITE(7,8)(XEIG(J,I),J=1,NCC)
  379 DO 381 KK=1,NSU
      DO 381 III=1,ITE
      IF(ITY(III).EQ.0) GO TO 381
      MA=0
      K=K+1
      NGK=NG(K)
      WRITE(6,47) KK,III
      DO 380 I=1,NGK
```

```
      J=NOM(I,K)
      WRITE(6,46) I,IGRT(I,K),B(I,K),(MN(MA+L,K),L=1,J)
  380 MA=MA+J
      WRITE(7, 4)  (B(I,K),I=1,NGK)
      WRITE(7,10) (IGRT(I,K),I=1,NGK)
  381 CONTINUE
      GO TO 222
  777 WRITE(6,51)
  222 CONTINUE
      CALL EXIT
      STOP
      END
      SUBROUTINE VARI(K)                                                SUB  1
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
C***********************************************************************
C*    THIS SUBROUTINE GENERATES VARIOUS VARIABLES FOR KTH SUBSTRUCTURE *
C***********************************************************************
      NCI=NIC(K)
      N1=NCB(K)
      NU1=NBW1(K)
      NU2=NBW2(K)
      NU3=NBW3(K)
      NB=NBJ(K)
      NJK=NJ(K)
      NC=NCI+N1
      MA=0
      RETURN
      END
      SUBROUTINE ELESTF(M5,III,I8,K,KK,M6,M7,I9,ISPSP,NN,J8,M8,IDV,GG)  SUB  2
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
   10 FORMAT(16I5)
   35 FORMAT('0',' EL NO  JP   JQ   JR   MP',8X,'L/SA',9X,'L1',12X,'M1'
     1,12X,'N1',12X,'L2',12X,'M2',12X,'N2'/' ')
   36 FORMAT(1X,5I5,7E15.4)
      WRITE(6,35)
      DO 700 M=M6,M7
      READ(5,10)          MM,JP,JQ,JR,MP(M,III)
      XL=X(JQ,K)-X(JP,K)
      YM=Y(JQ,K)-Y(JP,K)
      ZN=Z(JQ,K)-Z(JP,K)
      IF(III.GT.1) GO TO 500
      ID=3
C.... TRUSS ELEMENT STIFF. MATRIX.
      ELL(M, 1)=DSQRT(XL*XL+YM*YM+ZN*ZN)
```

```
      EL=1./ELL(M,III)
      CL=XL*EL
      CM=YM*EL
      CN=ZN*EL
      WRITE(6,36) M,JP,JQ,JR,MP(M,III),ELL(M,1),CL,CM,CN
      CON=E(IGRE(M,1),KK)*EL
      ESF(1,1)=CL
      ESF(1,2)=CM
      ESF(1,3)=CN
      B1(1,1)=CL*CL
      B1(2,2)=CM*CM
      B1(3,3)=CN*CN
      B1(1,2)=CL*CM
      B1(1,3)=CL*CN
      B1(2,3)=CM*CN
      IF(IDV.EQ.0) GO TO 600
      CONM=RO(IGRE(M,III))*ELL(M,III)/(6.0*GG)
      B2(1,1)=1.0
      IDM=1
      GO TO 599
  500 IF(III.GT.2) GO TO 515
      ID=9
C.... CST ELEMENT STIFF. MATRIX.
      BX=DSQRT(XL*XL+YM*YM+ZN*ZN)
      ESF(2,1)= XL/BX
      ESF(2,2)= YM/BX
      ESF(2,3)= ZN/BX
      XL=X(JR,K)-X(JP,K)
      YM=Y(JR,K)-Y(JP,K)
      ZN=Z(JR,K)-Z(JP,K)
      SX=XL*ESF(2,1)+YM*ESF(2,2)+ZN*ESF(2,3)
      XL=XL-SX*ESF(2,1)
      YM=YM-SX*ESF(2,2)
      ZN=ZN-SX*ESF(2,3)
      HX=DSQRT(XL*XL+YM*YM+ZN*ZN)
      ESF(1,1)=XL/HX
      ESF(1,2)=YM/HX
      ESF(1,3)=ZN/HX
      ELL(M, 2)=0.5*BX*HX
      WRITE(6,36) M,JP,JQ,JR,MP(M,III),ELL(M,2),((ESF(J,L),L=1,3),J=1,2)
      XNU=XNUU(IGRE(M, 2),KK)
      ETA=(1.0-XNU)*0.5
      CON=E(IGRE(M,2),KK)/((1.0-XNU*XNU)*2.0*BX*HX)
      BMS=BX-SX
      HH=HX*HX
      SZ=SX*SX
      BB=BX*BX
      BMSS=BMS*BMS
      SBMS=SX*BMS
      HBMS=HX*BMS
      BBMS=BX*BMS
      B1(1,1)=BMSS+ HH*ETA
      B1(1,2)=(XNU+ETA)*HBMS
      B1(1,3)=SBMS-HH*ETA
      B1(1,4)=-HBMS*XNU+HX*SX*ETA
      B1(1,5)=-BBMS
      B1(1,6)=-HX*BX*ETA
      B1(2,2)= HH+BMSS*ETA
      B1(2,3)= XNU*SX*HX-HBMS*ETA
      B1(2,4)=-HH+SBMS*ETA
      B1(2,5)=-BX*HX*XNU
```

```
      B1(2,6)=-BBMS*ETA
      B1(3,3)= SZ+HH*ETA
      B1(3,4)=-(XNU+ETA)*HX*SX
      B1(3,5)=-SX*BX
      B1(3,6)= HX*BX*ETA
      B1(4,4)= HH+SZ*ETA
      B1(4,5)= HX*BX*XNU
      B1(4,6)=-SX*BX*ETA
      B1(5,5)= BB
      B1(5,6)= 0.0
      B1(6,6)= BB*ETA
      DO 501 J=1,6
      DO 501 L=J,6
  501 B1(L,J)=B1(J,L)
      DO 502 J=1,6
      DO 502 L=1,9
  502 B2(J,L)=0.0
      JJ=0
      DO 504 LS=1,5,2
      LE=LS+1
      DO 504 J=1,3
      JJ=JJ+1
      DO 504 L=1,6
  504 B2(L,JJ)=B2(L,JJ)+ B1(L,LS)*  ESF(1,J)+     B1(L,LE)*   ESF(2,J)
      DO 503 J=1,9
      DO 503 L=J,9
  503 B1(J,L)=0.0
      JJ=0
      DO 506 LS=1,5,2
      LE=LS+1
      DO 506 J=1,3
      JJ=JJ+1
      DO 506 L=JJ,9
  506 B1(JJ,L)=B1(JJ,L)+B2(LS,L)*  ESF(1,J)+B2(LE,L)*   ESF(2,J)
      IF(IDV.EQ.0) GO TO 600
      CONM=RO(IGRE(M,III))*ELL(M,III)/(12.0*GG)
      B2(1,1)=1.0
      IDM=1
      GO TO 599
C.... SSP ELEMENT STIFF. MATRIX.
  515 ID=6
      SSPB=DABS(Z(JP,K)+Z(JQ,K))
      SSPA=DSQRT(XL*XL+YM*YM)
      ESF(1,1)=XL/SSPA
      ESF(1,2)=YM/SSPA
      ELL(M, 3)=0.5*SSPA*SSPB
      WRITE(6,36) M,JP,JQ,JR,MP(M,III),ELL(M,3),ESF(1,1),ESF(1,2)
      XNU=XNUU(IGRE(M, 3),KK)
      THETA=SSPA/SSPB
      CON=E(IGRE( M,3),KK)/(12.0*(1.0+XNU))
      DCM=ESF(1,2)
      DCL=ESF(1,1)
      DCLL=DCL*DCL
      DCLM=DCL*DCM
      DCMM=DCM*DCM
      Z1=2.0*(1.0+XNU)/THETA
      IF(ISPSP.NE.0)Z1=0.0
      Z2=3.0*THETA
      S11=Z1+Z2
      S13=-Z1+Z2
      S22=3.0/THETA
```

```
      B1(1,1)= S11*DCLL
      B1(1,2)= S11*DCLM
      B1(1,3)=-3.0*DCL
      B1(1,4)= S13*DCLL
      B1(1,5)= S13*DCLM
      B1(1,6)= 3.0*DCL
      B1(2,2)= S11*DCMM
      B1(2,3)=-3.0*DCM
      B1(2,4)= S13*DCLM
      B1(2,5)= S13*DCMM
      B1(2,6)= 3.0*DCM
      B1(3,3)= S22
      B1(3,4)=-3.0*DCL
      B1(3,5)=-3.0*DCM
      B1(3,6)=-S22
      B1(4,4)= S11*DCLL
      B1(4,5)= S11*DCLM
      B1(4,6)= 3.0*DCL
      B1(5,5)= S11*DCMM
      B1(5,6)= 3.0*DCM
      B1(6,6)= S22
      IF(IDV.EQ.0) GO TO 600
      CONM=RO(IGRE(M,III))*SSPB*SSPB/(6.0*GG)
      XM11=THETA/3.0+XNU*THETA/6.0+(THETA**3)/10.0+0.1*XNU*XNU/THETA
      XM12=-0.25*(THETA*THETA+XNU)
      XM13=THETA/6.0-XNU*THETA/6.0-(THETA**3)/10.0-0.1*XNU*XNU/THETA
      XM22=THETA
      B2(1,1)=DCLL*XM11
      B2(1,4)=DCLL*XM13
      B2(4,4)=B2(1,1)
      B2(1,2)=DCLM*XM11
      B2(1,5)=DCLM*XM13
      B2(2,4)=DCLM*XM13
      B2(4,5)=DCLM*XM11
      B2(1,3)=DCL*XM12
      B2(1,6)=DCL*XM12
      B2(3,4)=-DCL*XM12
      B2(4,6)=-DCL*XM12
      B2(2,2)=+DCMM*XM11
      B2(2,5)=DCMM*XM13
      B2(5,5)=+DCMM*XM11
      B2(2,3)=+DCM*XM12
      B2(2,6)=+DCM*XM12
      B2(3,5)=-DCM*XM12
      B2(5,6)=-DCM*XM12
      B2(3,3)=+XM22
      B2(3,6)=XM22*0.5
      B2(6,6)=+XM22
      IDM=6
  599 ICSSM(M,III)=M8
      DO 616 J=1,IDM
      DO 616 L=1,J
      M8=M8+1
  616 TCSM(M8)=CONM*B2(L,J)
  600 ICSS( M,III)=I8
      DO 516 J=1,ID
      DO 516 L=1,J
      I8=I8+1
  516 TRCSSP(I8)=CON*B1(L,J)
      L=NN*(JP-1)
      I=NN*(JQ-1)
```

```
      N=NN*(JR-1)
      DO 517 J=1,NN
      NA(J)=ND(L+J)
      NA(J+NN)=ND(I+J)
      IF(III.EQ.2) NA(J+ SN)=ND(N+J)
  517 CONTINUE
      ISAC( M,III)=I9
      IF(III.EQ.1) ID=6
      DO 519 J=1,ID
      I9=I9+1
      NNDC(I9)=NA(J)
  519 CONTINUE
      IF(III.GT.1) GO TO 800
      DO 400 I=1,3
      DO 400 J=1,I
  400 B1(I,J)=B1(J,I)
      DO 801 I=1,3
      H(I)=0.0
      DO 801 J=1,3
  801 H(I)=H(I)+ESF(1,J)*B1(J,I)
      INDC(M,III)=J8
      LLN(III)=NN
      DO 802 J=1,NN
      J8=J8+1
  802 STRESS(J8)=H(J)*CON
      GO TO 700
  800 IF(III.GT.2) GO TO 716
      LN=3
      LLN(III)=LN
C.... STRESS MATRIX FOR CST ELEMENTS.
      CON=CON*2.0
      B1(1,1)=-BMS
      B1(1,2)=-HX*XNU
      B1(1,3)=-SX
      B1(1,4)=-B1(1,2)
      B1(1,5)= BX
      B1(1,6)= 0.0
      B1(2,1)=-BMS*XNU
      B1(2,2)=-HX
      B1(2,3)=-SX*XNU
      B1(2,4)= HX
      B1(2,5)= BX*XNU
      B1(2,6)= 0.0
      B1(3,1)=-HX*ETA
      B1(3,2)=-BMS*ETA
      B1(3,3)= HX*ETA
      B1(3,4)=-SX*ETA
      B1(3,5)= 0.0
      B1(3,6)= BX*ETA
      DO 713 I=1,3
      DO 713 J=1,9
  713 B2(I,J)=0.0
      JJ=0
      DO 708 LS=1,5,2
      LE=LS+1
      DO 708 J=1,3
      JJ=JJ+1
      DO 708 I=1,3
  708 B2(I,JJ)=B2(I,JJ)+B1(I,LS)*  ESF(1,J)+B1(I,LE)*  ESF(2,J)
      GO TO 805
C.... STRESS MATRIX FOR SSP ELEMENTS.
```

```
 716 ET3=0.5/(1.0+XNU)
     CON=E(IGRE(M,III),KK)
     ET3=ET3*CON
     AX=1.0/SSPA
     BX=1.0/SSPB
     IF(ISPSP.EQ.1) GO TO 731
     B2(1,1)=-DCL*AX*CON
     B2(1,2)=-DCM*AX*CON
     B2(1,3)= 0.0
     B2(1,4)=-B2(1,1)
     B2(1,5)=-B2(1,2)
     B2(1,6)= 0.0
     J=2
 731 IF(ISPSP.NE.0)J=1
     B2(J,1)= DCL*ET3*BX
     B2(J,2)= DCM*ET3*BX
     B2(J,3)=-ET3*AX
     B2(J,4)= B2(J,1)
     B2(J,5)= B2(J,2)
     B2(J,6)=-B2(J,3)
     LN=J
     LLN(III)=J
     CON=1.0
 805 INDC(M,III)=J8
     LE=ITY(III)
     DO 336 I=1,LN
     DO 336 J=1,LE
     J8=J8+1
 336 STRESS(J8)=B2(I,J)*CON
 700 CONTINUE
     RETURN
     END
     SUBROUTINE STIFFM(N,K,NDC,I8,*,IO1)
     IMPLICIT REAL*8 (A-H,O-Z)
     INTEGER SIZE,BNC,SN
     COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
     COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
     COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
    1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
     COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
    1),NJ1( 9),NJ2( 9)
     COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
    1 156),TRCSSP(2809),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
    2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
    3 108, 2)
     COMMON/P3/EVEC(  1, 1),RRF( 7),RDLIM( 7),RSL( 7),RSU( 7),RLOAD( 7)
    1,REDUC( 90),NDOF( 7),NDM( 90),NBDAM( 6, 6),KIIDAM( 3, 7)
     COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
    1 6)
     COMMON/R4/IIL( 50, 3),KLC( 50),IOK( 3),NO( 1)
     COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
    1,SS( 51)
     COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
     COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
    1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
     COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
     COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
    1 156),Y( 108, 3),NZC( 24, 3)
     COMMON/C1/XEIG( 72, 2),YXEIG( 72, 2),WS( 2),DM(  1,  1),IETA( 7)
    $/C3/ QQK( 2, 2),QQM( 2, 2),QA( 2, 2)
************************************************************************
```

```
C*     DESCRIPTION OF VARIABLES                                                *
C*          U      - STORES (E(I)*B(I))/L(I)          ( NO. OF MENBERS   )    *
C*          DKI      STORES  KII  IN BANDED DECOMPOSED FORM.                   *
C*          DPZ      STORES  KBI  IN FULL                                      *
C*          C        STORES  KB   FOR WHOLE STR IN BANDED FORM                 *
C*************************************************************************************
      I0=0
      KK=0
      IDC=NDC-1
      INDEX=I01
      DO 999 K=1,NSU
C
      CALL VARI(K)
      IOK(K)=0
      DO 12 I=1,N1
      DO 12 J=1,N1
   12 A2(I,J)=0.D0
      IF(NCI.EQ.0) GO TO 15
      DO 14 I=1,NCI
      NA(I)=0
      DO 11 L=1,NLC
   11 BE(I,L)=PI(I,L,K)*RLOAD(NDC)
      DO 13 J=1,NU3
   13 D(I,J)=0.D0
      DO 14 J=1,N1
      DPZ(I,J)=0.D0
   14 DS(I,J)=0.D0
   15 DO 29 III=1,3
      IF(ITY(III).EQ.0) GO TO 29
      KK=KK+1
      M6=MEB(KK)
      M7=MEF(KK)
      DO 16 I4=M6,M7
   16 ND(I4)=0
      IF(IDC.EQ.0) GO TO 18
      NDO=NBDAM(KK,IDC)
      IF(NDO.EQ.0) GO TO 18
      DO 17 I4=1,NDC
      I8=I8+1
      ND(NDM(I8))=I8
 17   CONTINUE
   18 DO 28 I4=M6,M7
      Z1=1.0
      IF(ND(I4).NE.0)GO TO 202
      XX=B(IGRE(I4,III),KK)*Z1
      BR(I4,III)=XX
      GO TO 203
  202 Z1=1.D0-REDUC(ND(I4))
      XX=B(IGRE(I4,III),KK)*Z1
      BR(I4,III)=XX
  203 IF(XX.EQ.0.0)GO TO 28
C
      CALL RECALL(III,LE,LS,LF,INDEX,I4,XX)
      IF(NCI.EQ.0) GO TO 25
      DO 24 J=LS,LE
      IJ=NNDC(LF+J)
      IF(IJ.EQ.0) GO TO 24
      IF(IJ.GT.N1) GO TO 21
      DO 20 L=LS,LE
      IL=NNDC(LF+L)
      IF(IL.EQ.0) GO TO 20
```

```
      IF(IL.GT.N1.OR.IL.LT.IJ) GO TO 20
      IK=IL-IJ+1
      IF(IK.GT.NU2) GO TO 20
      A2(IJ,IL)=A2(IJ,IL)+ESF(J,L)
   20 CONTINUE
      GO TO 24
   21 IJ=IJ-N1
      DO 23 L=LS,LE
      IL=NNDC(LF+L)
      IF(IL.EQ.0) GO TO 23
      IF(IL.GT.N1) GO TO 22
      DS(IJ,IL)=  DS(IJ,IL)- ESF(J,L)
      DPZ(IJ,IL)=-DS(IJ,IL)
      GO TO 23
   22 IF(KIIDAM(K,NDC).EQ.0) GO TO 23
      IL=IL-N1
      IF(IL.LT.IJ) GO TO 23
      IK=IL-IJ+1
      IF(IK.GT.NU3) GO TO 23
      D(IJ,IK)=  D(IJ,IK)+ESF(J,L)
   23 CONTINUE
   24 CONTINUE
      GO TO 28
   25 DO 27 J=LS,LE
      IJ=NNDC(LF+J)
      IF(IJ.EQ.0) GO TO 27
      DO 26 L=LS,LE
      IL=NNDC(LF+L)
      IF(IL.EQ. 0.OR.IL.LT.IJ) GO TO 26
      IK=IL-IJ+1
      IF(IK.GT.NU2) GO TO 26
      A2(IJ,IL)=A2(IJ,IL)+ESF(J,L)
   26 CONTINUE
   27 CONTINUE
   28 CONTINUE
   29 CONTINUE
      IF(NCI.EQ.0) GO TO 126
      IF(NDC.GT.1) GO TO 35
      N=0
C
      CALL DECUPP(N,NU3,NCI,&444)
C     D   CONTAINS DECOMPOSED KII.
C     WRITE(6,31) ((D(I,J),J=1,NU3),I=1,NCI)
C  31 FORMAT(3X,6E15.5)
      GO TO 33
  444 RETURN 1
   33 KIIUBW(K)=IQ
      DO 34 J=1,NU3
      IQ=IQ+1
      DO 34 I=1,NCI
      DPB(I,IQ)= D(I,J)
   34 DKI(I,IQ)= D(I,J)
      GO TO 42
   35 IF(KIIDAM(K,NDC).EQ.0) GO TO 39
      DO 36 I=1,NCI
      IF(D(I,1).NE.0.0) GO TO 36
      NA(I)=1
      D(I,1)=1.0
   36 CONTINUE
      N=0
C
```

```
      CALL DECUPP(N,NU3,NCI,&444)
      IQ=KIIUBW(K)
      DO 37 J=1,NU3
      IQ=IQ+1
      DO 37 I=1,NCI
   37 DPB(I,IQ)= D(I,J)
      GO TO 42
   39 IQ=KIIUBW(K)
      DO 40 J=1,NU3
      IQ=IQ+1
      DO 40 I=1,NCI
      D(I,J)=DKI(I,IQ)
   40 DPB(I,IQ)=DKI(I,IQ)
   42 DO 43 L=1,NLC
      J=L+N1
      DO 43 I=1,NCI
      Z1=1.0
      IF(NA(I).EQ.1) Z1=0.0
      DS(I,J)=BE(I,L)*Z1
   43 BE(I,L)=DS(I,J)
C
      CALL SOLDUP(J, NU3,NCI)
C.... DS CONTAINS Q=-KII**-1*KIB AND KII**-1*PI*RLOAD
      DO 49 I=1,NCI
      DO 48 L=1,N1
   48 Q(I,L,K)=DS(I,L)
      DO 49 L=1,NLC
      J=L+N1
   49 ZI(I,L,K)=DS(I,J)
C     WRITE(6,39) K,((Q(I,J,K),I=1,NCI),J=1,N1)
C  39 FORMAT(//3X,I2,' MATRIX Q'/(3X,4E15.5))
C.... GENERATION OF KB FOR WHOLE STRUCTURE IN BANDED FORM .
      MCI=NU1-N1-1
      DO 124 I=1,N1
      MCI=MCI+1
      IF(MCI.GT.NCI) MCI=NCI
      DO 124 J=I,N1
      DO 124 L=1,MCI
  124 A2(I,J)=A2(I,J)+DPZ(L,I)*Q(L,J,K)
C.... GENERATION OF RB   EFFECTIVE BOUND FORCE VECTOR IN MATRIX ZB.
      DO 125 I=1,N1
      L1=NZC( I,K)
      DO 125 L=1,NLC
      DO 125 J=1,NCI
  125 ZB(L1,L)=ZB(L1,L)+Q(J,I,K)*BE(J,L)
  126 DO 127 I=1,N1
      L1=NZC(I,K)
      DO 127 J=I,N1
      L2=NZC(J,K)
      IF(L2.LT.L1) GO TO 127
      L3=L2-L1+1
      IF(L3.GT.NBW) GO TO 127
      C(L1,L3)=C(L1,L3)+A2(I,J)
  127 CONTINUE
  999 CONTINUE
C     WRITE(6,1004) ((C(I,J),J=1,NBW),I=1,BNC)
      IF(IDC.EQ.0) RETURN
      DO 129 I=1,BNC
      IF(C(I,1).NE.0.0) GO TO 129
      C(I,1)=1.0
      DO 128 J=1,NLC
```

```
128 ZB(I,J)=0.0
129 CONTINUE
    RETURN
    END
    SUBROUTINE RECALL(III,LE,LS,LF,INDEX,MI,XX)                          SUB  4
    IMPLICIT REAL*8 (A-H,O-Z)
    INTEGER SIZE,BNC,SN
    COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
    COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
   1),NJ1( 9),NJ2( 9)
    COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
   1 156),TRCSSP(2808),XCUST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
   2, 2),IGRT( 14, 6),IGRF( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
   3 108, 2)
    COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
   1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
    NN=SN/2
    LE=ITY(III)
    IF(III.EQ.1) LE=3
    LS=1
    LF=ICSS(MI,III)
    IF(INDEX.EQ.2)GO TO 702
    DO 701 J=LS,LE
    DO 701 I=1,J
    LF=LF+1
    ESF(I,J)=TRCSSP(LF)*XX
701 ESF(J,I)=ESF(I,J)
    IF(III.GT.1) GO TO 702
    DO 401 JO=1,3
    DO 401 IO=1,3
    ESF(JO,IO+3)=-ESF(JO,IO)
401 ESF(JO+3,IO+3)=ESF(JO,IO)
    DO 402 I=1,6
    DO 402 J=1,I
402 ESF(I,J)=ESF(J,I)
702 IF(INDEX.EQ.0) GO TO 170
    IF(III.GT.1) GO TO 150
    Y1=TCSM(ICSSM(MI,III)+1)*XX
    Y2=Y1+Y1
    DO 60 I=1,SN
    DO 60 J=1,SN
    A=0.0
    IF(I.EQ.J) A=Y2
    IX=I-J
    IF((IX+NN)*(IX-NN).EQ.0) A=Y1
 60 B1(I,J)=A
    GO TO 170
150 IF(III.GT.2) GO TO 160
    Y1=TCSM(ICSSM(MI,III)+1)*XX
    Y2=Y1+Y1
    DO 59 I=1,9
    DO 59 J=1,9
59  B1(I,J)=0.0
    DO 61 I=1,3
    DO 61 J=1,3
    A=Y1
    IF(I.EQ.J)A=Y2
61  B1(I,J)=A
    LI=3
    LJX=3
    DO 63 II=1,2
```

```
      DO 62 I=1,3
      LI=LI+1
      LJ=LJX
      DO 62 J=1,3
      LJ=LJ+1
   62 B1(LI,LJ)=B1(I,J)
      LI=6
   63 LJX=6
      GO TO 170
  160 L=ICSSM(MI,III)
      DO 65 J=1,6
      DO 65 I=1,J
      L=L+1
      B1(I,J)=TCSM(L)*XX
   65 B1(J,I)=B1(I,J)
  170 LE=ITY(III)
      LS=1
      LF=ISAC(MI,III)
      RETURN
      END
      SUBROUTINE DECUPP(M,IU,N,*)                                       SUB   5
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
C*********************************************************************
C*    DECOMPOSE A SYMMETRIC MATRIX                                   *
C*    UPPER BANDED MATRIX IS ASSUMED                                 *
C*    UPPER DECOMPOSED MATRIX IS STORED IN THE ORIGINAL POSITION     *
C*    ORIGINAL MATRIX IS DESTROYED                                   *
C*    N=BNC      IU=BUBW                                             *
C*********************************************************************
   13 FORMAT('1',30X,'SINGULAR MATRIX',I8)
      DO 60 I=1,N
      IP=N-I+1
      L=I-1
      IF(IU.LT.IP) IP=IU
      DO 60 J=1,IP
      SUM=D(I,J)
      IF(I.EQ.1) GO TO 40
      IQ=IU-J
      IF(L.LT.IQ) IQ=L
      IF(IQ.EQ.0) GO TO 40
      DO 30 K=1,IQ
      MZ=I-K
   30 SUM=SUM-D(MZ,K+1)*D(MZ,K+J)
   40 IF(J.NE.1) GO TO 50
      IF(SUM.LE.0.) GO TO 100
      TEMP=DSQRT(SUM)
      TEMP=1.0/TEMP
      D(I,J)=TEMP
      GO TO 60
   50 D(I,J)=SUM*TEMP
   60 CONTINUE
      GO TO 91
  100 CONTINUE
      M=I
      WRITE(6,13) I
      RETURN 1
   91 RETURN
      END
      SUBROUTINE SOLDUP(NL,IU,N)                                        SUB   6
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
C***********************************************************************
C*    SOLUTION OF SIMULTANEOUS EQUATIONS BY DECOMPOSING THE MATRIX     *
C*    UPPER TRIANGULAR BANDED MATRIX IS ASSUMED                        *
C*    SOLVE BY FORWARD AND BACKWARD SUBSTITUTIONS                      *
C*    DS IS THE RHS MATRIX                                             *
C*    DS CONTAINS THE SOLUTION AT THE END                              *
C*    DS IS NOT SAVED                                                  *
C*    FORWARD SUBSTITUTION                                             *
C*    N=BNC       IU=BUBW                                              *
C***********************************************************************
C  60 FORMAT(2X,' BOUNDARY DISPLACEMENTS IN OVERALL SYSTEM'/(3X,4E15.5))
      DO 20 I=1,NL
   20 DS(1,I)=DS(1,I)*D(1,1)
      DO 10 I=2,N
      J=I-IU+1
      IF((I+1).LE.IU) J=1
      IJ=I-1
      DO 25 II=1,NL
      DO 15 K=J,IJ
      LS=I-K+1
   15 DS(I,II)=DS(I,II)-D(K,LS)*DS(K,II)
   25 DS(I,II)=DS(I,II)*D(I,1)
   10 CONTINUE
C.... BACKWARD SUBSTITUTION
      DO 30 I=1,NL
   30 DS(N,I)=DS(N,I)*D(N,1)
      L=N-1
      DO 90 II=1,L
      I=N-II
      JI=I-1
      J=JI+IU
      IF(J.GT.N) J=N
      IJ=I+1
      DO 85 M=1,NL
      DO 95 K=IJ,J
   95 DS(I,M)=DS(I,M)-D(I,K-JI)*DS(K,M)
   85 DS(I,M)=DS(I,M)*D(I,1)
   90 CONTINUE
C     WRITE(6,60) ((DS(I,J),I=1,N),J=1,NL)
      RETURN
      END
      SUBROUTINE MEVEC(NN,NCC,LDC,I8,IA)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P3/EVEC(  1, 1),RRF( 7),RDLIM( 7),RSL( 7),RSU( 7),RLOAD( 7)
     1,REDUC( 90),NDOF( 7),NDM( 90),NBDAM( 6, 6),KIIDAM( 3, 7)
      COMMON/P5/YK(  1),YM(  1),SK(  1),SM(  1),EY(  1),SS(  1)
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
```

```
      1Z( 51, 3),DZE( 60),MP( 108, 2),ND( 216)
       COMMON/A6/DPZ( 50, 30),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
      1 156),Y( 108, 3),NZC( 24, 3)
       COMMON/C1/XEIG( 72, 2),YXEIG( 72, 2),WS( 2),DM( 1, 1),IET( 7)
       IDC=LDC-1
       NCX=BNC
       KK=0
       IO12=2
       INDEX=IO12
       DO 113 I=1,NCC
       DO 113 JJ=1,IA
  113  YXEIG(I,JJ)=0.D0
       DO 108 K=1,NSU
C
       CALL VARI(K)
       DO 80004 III=1,3
       IF(ITY(III).EQ.0)GO TO 80004
       KK=KK+1
       M6=MEB(KK)
       M7=MEF(KK)
       DO 16 I4=M6,M7
   16  ND(I4)=0
       IF(IDC.EQ.0) GO TO 18
       NDO=NBDAM(KK,IDC)
       IF(NDO.EQ.0) GO TO 18
       DO 17 I4=1,NDO
       I8=I8+1
   17  ND(NDM(I8))=I8
   18  DO 107 I4=M6,M7
       Z1=1.0
       IF(ND(I4).NE.0) Z1=1.0-REDUC(ND(I4))
       XX=B(IGRE(I4,III),KK)*Z1
       BR(I4,III)=XX
       IF(XX.EQ.0.0) GO TO 107
C
       CALL RECALL(III,LE,LS,LF,INDEX,I4,XX)
       DO 106 I=LS,LE
       IJ=NNDC(LF+I)
       IF(IJ.EQ.0) GO TO 106
       IF(IJ.LE.N1) I5=NZC(IJ,K)
       IF(IJ.GT.N1) I5=NCX+IJ-N1
       DO 105 J=LS,LE
       IL=NNDC(LF+J)
       IF(IL.EQ.0) GO TO 105
       IF(IL.LE.N1) I6=NZC(IL,K)
       IF(IL.GT.N1) I6=NCX+IL-N1
       DO 109 JJ=1,IA
  109  YXEIG(I5,JJ)=YXEIG(I5,JJ)+B1(I,J)*XEIG(I6,JJ)
  105  CONTINUE
  106  CONTINUE
  107  CONTINUE
80004  CONTINUE
       NCX=NCX+NCI
  108  CONTINUE
       RETURN
       END
       SUBROUTINE DEFREQ(FREQ,RFF,NN,FDEN,NCC)
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER SIZE,BNC,SN
       COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
       COMMON/V1/N1,NCI,NWK,NJK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),EU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P5/YK(  1),YM(  1),SK(  1),SM(  1),EY(  1),SG(  1)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
      COMMON/C1/XEIG( 72, 2),YXEIG( 72, 2),WS( 2),DM(  1,  1),IET( 7)
C*********************************************************************
C*    GENERATES SENSITIVITY VECTOR   H( NV ) FOR FREQUENCY CONSTRAINT  *
C*         FREQ - NATURE FREQUENCY OF THE STRUCTURE                    *
C*         RFF  - FREQUENCY LIMIT                                      *
C*         SK   - EIGEN VECTOR      (NCC)                              *
C*         W    - STORES D( K*Y - F*M*Y )/D B(I)                       *
C*********************************************************************
      LD=0
      INDEX=1
      NCX=BNC
      SUM=FDEN*RFF
      DO 106 K=1,NSU
C
      CALL VARI(K)
      DO 80004 III=1,3
      IF(ITY(III).EQ.0)GO TO 80004
      MA=0
      LD=LD+1
      NGK=NG(LD)
      DO 100 KK=1,NGK
      NJJ=NOM(KK,LD)
      MV=IGRT(KK,LD)
      IF(MV.EQ.0) GO TO 170
      DO 102 J=1,NCC
  102 W(J)=0.D0
      DO 80 IM=1,NJJ
      MA=MA+1
      MI=MN(MA,LD)
      XX=BE(MI,III)
      IF(XX.EQ.0.0) GO TO 80
      XX=1.0
C
      CALL RECALL(III,LE,LS,LF,INDEX,MI,XX)
      DO 70 I=LS,LE
      IJ=NNDC(LF+I)
      IF(IJ.EQ.0) GO TO 70
      IF(IJ.LE.N1) I4=NZC(IJ,K)
      IF(IJ.GT.N1) I4=NCX+IJ-N1
      DO 60 J=LS,LE
      IL=NNDC(LF+J)
      IF(IL.EQ.0) GO TO 60
      IF(IL.LE.N1) I5=NZC(IL,K)
      IF(IL.GT.N1) I5=NCX+IL-N1
      W(I4)=W(I4)+(ESF(I,J)-FREQ*B1(I,J))*XEIG(I5,1)
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
```

```
      H(MV)=0.D0
      DO 90 J=1,NCC
   90 H(MV)= H(MV)+W(J)*XEIG(J,1)
      H(MV)=  H(MV)/SUM
      GO TO 100
  170 MA=MA+NJJ
  100 CONTINUE
80004 CONTINUE
      NCX=NCX+NCI
  106 CONTINUE
      RETURN
      END
      SUBROUTINE ZBZIEF(IDC,ISPSP,IPS,IPD)                              SUB 10
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISO,IQ1
      COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P3/EVEC(  1, 1),RRF( 7),RDLIM( 7),RSL( 7),RSU( 7),RLOAD( 7)
     1,REDUC( 90),NDOF( 7),NDM( 90),NBDAM( 6, 6),KIIDAM( 3, 7)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),L( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C*    THIS SUBROUTINE COMPUTES NODAL DISPLACEMENTS
C     AND MEMBER FORCES/VON MISES EQUIVALENT STRESS.
   35 FORMAT(45X,'FORCE  MATRIX FOR TRUSS ELEMENTS'/45X,'ELEMENT FORCE (
     * + IS COMP.)')
   36 FORMAT(45X,'STRESS MATRIX FOR   CST ELEMENTS'/45X,'ELEMENT SIGMA-X
     *, SIGMA-Y, TAU-XY, VON MISES STRESS')
   37 FORMAT(45X,'STRESS MATRIX FOR SHEAR ELEMENTS'/45X,'ELEMENT TAU-XY,
     * VON MISES STRESS')
   38 FORMAT(45X,'STRESS MATRIX FOR   SSP ELEMENTS'/45X,'ELEMENT SIGMA-X
     *, TAU-XY, VON MISES STRESS')
   39 FORMAT(45X,I5,4E15.5)
   40 FORMAT(45X,'LOADING CONDITION= ',I5)
   42 FORMAT(3X,I5,4X,6E15.5)
   43 FORMAT('0',' ***NODAL DISPLACEMENTS FOR K=',I2/'   1ST ZB THEN ZI
     *FOR ALL NLC.')
   58 FORMAT(/' ',' ***FORCE/VON MISES STRESS FOR K=',I2,', III=',I2,',
     *ITY(III)=',I2,', LDC=',I2)
      LDC=IDC-1
      DO 303 I=1,BNC
      DO 302 L=1,NLC
  302 DS(I,L)=ZB(I,L)
      DO 303 J=1,NBW
  303 C(I,J)=D(I,J)
      CALL SOLDUP(NLC,NBW,BNC)
      DO 305 J=1,NLC
```

```
      DO 305 I=1,BNC
  305 ZB(I,J)=DS(I,J)
      LL=0
      DO 777 K=1,NSU
      CALL VARI(K)
      DO 316 J=1,N1
      L1=NZC(J,K)
      DO 316 L=1,NLC
  316 ZZ(J,L)=ZB(L1,L)
      IF(NCI.EQ.0) GO TO 321
      DO 319 L=1,NLC
      DO 318 J=1,N1
      DO 318 I=1,NCI
  318 ZI(I,L,K)=ZI(I,L,K)+Q(I,J,K)*ZZ(J,L)
      DO 319 I=1,NCI
  319 ZZ(I+N1,L)=ZI(I,L,K)
      IF(IPD.EQ.0)GO TO 357
      WRITE (6,43)K
      DO 320 I=1,NC
  320 WRITE (6,42)I,(ZZ(I,J),J=1,NLC)
  321 IF(IPD.EQ.0)GO TO 357
      WRITE(6,43)K
      DO 358 J=1,N1
  358 WRITE(6,42)J,(ZZ(J,JJ),JJ=1,NLC)
  357 DO 777 III=1,3
      WRITE (6,58)K,III,ITY(III),LDC
      IF(ITY(III).EQ.0) GO TO 777
      MA=0
      LE=ITY(III)
      LN=LLN(III)
      NX=LN+1
      LL=LL+1
      M6=MEB(LL)
      M7=MEF(LL)
      IF(III.GT.1) GO TO 326
      DO 325 I4=M6,M7
      DO 325 L=1,NLC
  325 TRSF(I4,L)=0.0
      GO TO 330
  326 IF(III.GT.2) GO TO 328
      DO 327 I4=M6,M7
      DO 327 L=1,NLC
      DO 327 J=1,NX
  327 CSTF(I4,L,J)=0.0
      GO TO 330
  328 DO 329 I4=M6,M7
      DO 329 L=1,NLC
      DO 329 J=1,NX
  329 SSPF(I4,L,J)=0.0
  330 DO 666 I4=M6,M7
      BX=BR(I4,III)
      IF(BX.EQ.0.0) GO TO 666
      LF=ISAC(I4,III)
      IF(III.GT.1) GO TO 334
      L=INDC(I4,III)
      DO 331 J=1,LN
      L=L+1
      VV(J)=-STRESS(L)
  331 VV(J+3)=-VV(J)
      DO 333 J=1,LE
      IJ=NNDC(LF+J)
```

```
      IF(IJ.EQ.0) GO TO 333
      XB=VV(J)*BX
      DO 332 L=1,NLC
  332 TRSF(I4,L)=TRSF(I4,L)+ZZ(IJ,L)*XB
  333 CONTINUE
      GO TO 666
  334 L=INDC(I4,III)
      DO 335 I=1,LN
      DO 335 J=1,LE
      L=L+1
  335 B2(I,J)=STRESS(L)
      DO 340 L=1,NLC
      DO 336 I=1,LN
      T(I)=0.0
      DO 336 J=1,LE
      IL=NNDC(LF+J)
      IF(IL.EQ.0) GO TO 336
      T(I)=T(I)+B2(I,J)*ZZ(IL,L)
  336 CONTINUE
      IF(III.EQ.3) GO TO 338
      VON=DSQRT(T(1)*T(1)+T(2)*T(2)-T(1)*T(2)+3.0*T(3)*T(3))
      DO 337 I=1,LN
  337 CSTF(I4,L,I)=T(I)
      CSTF(I4,L,LN+1)=VON
      GO TO 340
  338 IF(ISPSP.EQ.0)GO TO 3338
      VON=DABS(T(1))
      GO TO 3339
 3338 VON=DSQRT(T(1)*T(1)+3.0*T(2)*T(2))
 3339 DO 339 I=1,LN
  339 SSPF(I4,L,I)=T(I)
      SSPF(I4,L,LN+1)=VON
  340 CONTINUE
  666 CONTINUE
      IF(IPS.EQ.0) GO TO 777
      IF(III.GT.1) GO TO 350
      WRITE (6,35)
      DO 349 L=1,NLC
      WRITE (6,40)L
      DO 349 M=M6,M7
  349 WRITE (6,39)M,TRSF(M,L)
      GO TO 777
  350 IF(III.GT.2) GO TO 352
      WRITE (6,36)
      DO 351 L=1,NLC
      WRITE (6,40)L
      DO 351 M=M6,M7
  351 WRITE (6,39) M,(CSTF(M,L,I),I=1,NX)
      GO TO 777
  352 IF(ISPSP.EQ.0) GO TO 353
      WRITE (6,37)
      GO TO 354
  353 WRITE (6,38)
  354 DO 355 L=1,NLC
      WRITE (6,40)L
      DO 355 M=M6,M7
  355 WRITE (6,39) M,(SSPF(M,L,I),I=1,NX)
  777 CONTINUE
      RETURN
      END
      SUBROUTINE CONST(IDC,IBUK,IDIS,IBDIS,NSD,EP,MV,IBU,IV,IPC,NTL,IFS,SUB 11
```

```
     1ISPSP,NDAM)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,NI1,ISQ,IQ1
      COMMON/V2/NIC( 3),IW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P3/EVEC(  1, 1),RRF( 7),RDLIM( 7),RSL( 7),RSU( 7),RLOAD( 7)
     1,REDUC( 90),NDOF( 7),NDM( 90),NBDAM( 6, 6),KIIDAM( 3, 7)
      COMMON/P4/INF( 50, 8),NGV( 14, 6),IND( 50),NDISP( 72)
      COMMON/R1/BL( 14, 6),DLIB( 36)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/R4/IIL( 50, 3),KLC( 50),IOK( 3),NO( 1)
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
   80 FORMAT(//'0',' CONSTRAINT VIOLATIONS FOR IDC=',I5)
      IV=0
      LDC=IDC-1
      LL=0
      IF(IPC.EQ.0) GO TO 199
      WRITE (6,80) LDC
  199 DO 200 I=1,NSD
      NA(I)=0
      DO 200 J=1,BNC
  200 A2(J,I)=0.0
      INN=0
      IF(IBDIS.EQ.0) GO TO 84
      DO 168 I=1,BNC
      DZE(I)=DLIB(I)
      DO 168 L=1,NLC
  168 ZZ(I,L)=ZB(I,L)
      K=-1
      I6=0
      NNN=BNC
      CALL ABSMAX(K,NNN,INN,I6,NSD,IV,LDC,IPC,EP,NDAM)
      IF(SIZE.LT.(NSD-NDAM-1))GO TO 84
      WRITE(6,53) SIZE,NSD,NTL
      RETURN
   84 INN=BNC
      DO 999 K=1,NSU
      CALL VARI(K)
      I6=0
      IF(NCI.EQ.0) GO TO 202
      DO 201 J=1,NSD
      DO 201 I=1,NCI
  201 DS(I,J)=0.0
  202 DO 888 III=1,3
```

```
      IF(ITY(III).EQ.0) GO TO 888
      IF(IPC.EQ.0) GO TO 203
      WRITE(6,58) K,III
  203 MA=0
      LL=LL+1
      NGK=NG(LL)
      LE=ITY(III)
      LN=LLN(III)
      NX=LN+1
      DO 777 KK=1,NGK
      NJJ=NOM(KK,LL)
      T(8)=SL(KK,LL)/RSL(IDC)
      IF(III.EQ.1) T(9)=SU(KK,LL)/RSU(IDC)
      DO 444 N=1,NJJ
      MA=MA+1
      M=MN(MA,LL)
      ND(M)=0
      R=BR(M,III)
      IF(R.EQ.0.0) GO TO 444
      R=1.0/R
      DO 333 L=1,NLC
      IF(III.GT.1) GO TO 207
      T(4)=TRSF(M,L)*R
      IF(T(4).LT.0.) GO TO 205
      T(5)=T(8)
      IF(IBUK.EQ.0) GO TO 206
      BUC=E(KK,LL)/(ELL(M,1)**2)
      T(2)=BUC*R
      IF(T(2).LE.T(5)) GO TO 206
      T(5)=T(2)
      ND(M)=1
      GO TO 206
  205 T(5)=T(9)
  206 T(6)=T(4)*T(5)
      GO TO 208
  207 IF(III.EQ.2) VON=CSTF(M,L,NX)
      IF(III.EQ.3) VON=SSPF(M,L,NX)
      IF(VON.EQ.0.D0)GO TO 444
      T(5)=0.5*T(8)/VON
      T(6)=VON*T(8)
  208 IF(N.GT.1.OR.L.GT.1) GO TO 209
      YM=T(5)
      XL=T(6)
      IJ=M
      LC=L
      GO TO 333
  209 IF(XL.GE.T(6)) GO TO 333
      YM=T(5)
      XL=T(6)
      IJ=M
      LC=L
  333 CONTINUE
  444 CONTINUE
      IF(NTL.GT.IFS) GO TO 210
      IF(IGRT(KK,LL).EQ.0) GO TO 777
      BNEW=XL*B(KK,LL)
      IF(BNEW.LT.BL(KK,LL)) BNEW=BL(KK,LL)
      IF(BNEW.GT.BU(KK,LL)) BNEW=BU(KK,LL)
      B(KK,LL)=BNEW
      GO TO 777
  210 IF((XL+EP).LT.1.) GO TO 777
```

```
      IF(NGV(KK,LL).NE.0) GO TO 211
      SIZE=SIZE+1
      IF(SIZE.GT.(NSD-NDAM-1))GO TO 214
      NGV(KK,LL)=SIZE
      GO TO 212
211   PQX=1.0-DLPH(NGV(KK,LL))
      IF(XL.LE.PQX)GO TO 777
 212  DLPH(NGV(KK,LL))=1.-XL
      DLP(NGV(KK,LL))=DABS(DLPH(NGV(KK,LL)))
      KLC(NGV(KK,LL))=LC
      IV=IV+1
      NA(IV)=0
      INO(IV)=NGV(KK,LL)
      IGR=IGRT(KK,LL)
      IBUC=0
      IF(IGR.EQ.0) GO TO 213
      SS(IGR)=1.0
      IF(ND(IJ).EQ.0) GO TO 213
      BUC=E(KK,LL)/(ELL(IJ,1)**2)
      T(2)=BUC/BR(IJ,III)
      IF(T(2).LE.T(8)) GO TO 213
      IBUC=1
                BE(IV,1)=-XL/BR(IJ,III)
                NA(IV)=IGR
 213  INF(NGV(KK,LL),1)=IJ
      INF(NGV(KK,LL),2)=K
      INF(NGV(KK,LL),3)=III
      INF(NGV(KK,LL),4)=IGRT(KK,LL)
      INF(NGV(KK,LL),5)=LDC
      INF(NGV(KK,LL),6)=IV
      INF(NGV(KK,LL),7)=LC
      INF(NGV(KK,LL),8)=IBUC
      NO(LC)=1
      IF(MP(IJ,III).EQ.-1) GO TO 710
      I6=I6+1
      IIL(I6,K)=IV
 710  LF=ISAC(IJ,III)
      L=INDC(IJ,III)
      IF(III.GT.1) GO TO 332
      DO 331 J=1,LN
      L=L+1
      VV(J)=-STRESS(L)
 331  VV(J+3)=-VV(J)
      GO TO 336
 332  DO 334 I=1,LN
      DO 334 J=1,LE
      L=L+1
 334  B3(I,J)=STRESS(L)
      IF(III.EQ.3) GO TO 335
      X1=2.0*CSTF(IJ,LC,1)-CSTF(IJ,LC,2)
      X2=2.0*CSTF(IJ,LC,2)-CSTF(IJ,LC,1)
      X3=6.0*CSTF(IJ,LC,3)
      GO TO 336
 335  X1=2.0*SSPF(IJ,LC,1)
      X2=6.0*SSPF(IJ,LC,2)
 336  DO 714 J=1,LE
      LJ=NNDC(LF+J)
      IF(LJ.EQ.0) GO TO 714
      IF(III.EQ.1) R=YM*VV(J)
      IF(III.EQ.2) R=YM*(X1*B3(1,J)+X2*B3(2,J)+X3*B3(3,J))
      IF(III.EQ.3) R=YM*(X1*B3(1,J)+X2*B3(2,J))
```

```
      IF(III.EQ.3.AND.ISPSP.NE.0)R=YM*(X1*B3(1,J))
      IF(LJ.GT.N1) GO TO 711
      L1=NZC(LJ,K)
      A2(L1,IV)=A2(L1,IV)+R
      GO TO 714
  711 IR=LJ-N1
      DS(IR,I6)=DS(IR,I6)+R
      DO 712 I=1,N1
      L1=NZC( I,K)
  712 A2(L1,IV)=A2(L1,IV)+R*Q(IR,I,K)
  714 CONTINUE
      IF(IPC.EQ.0) GO TO 777
      ISIZ=NGV(KK,LL)
      WRITE(6,57)ISIZ,IV,I6,IJ,MP(IJ,III),LC,XL,DLPH(ISIZ)
  777 CONTINUE
  888 CONTINUE
      IF(NTL.LE.IFS)GO TO 999
      GO TO 557
  214 SIZE=SIZE-1
      WRITE(6,53)   SIZE,NSD,NTL
      GO TO 600
  557 IF( IDIS.EQ.0.OR.NCI.EQ.0) GO TO 600
      DO 125 I=1,NCI
      DZE(I)=DLIM(I,K)
      DO 125 J=1,NLC
  125 ZZ(I,J)=ZI(I,J,K)
      NNN=NCI
      CALL ABSMAX(K,NNN,INN,I6,NSD,IV,LDC,IPC,EP,NDAM)
  600 IOK(K)=I6
      IF(I6.EQ.0) GO TO 998
      IU=KIIUBW(K)
      DO 85 J=1,NU3
      IU=IU+1
      DO 85 I=1,NCI
   85 D(I,J)=DPB(I,IU)
      CALL SOLDUP(I6,NU3,NCI)
      DO 87 J=1,I6
      I7=IIL(J,K)
      DO 87 I=1,NCI
   87 DPZ(I,I7)=DS(I,J)
C     WRITE(6,311) ((DS(I,J),I=1,NCI),J=1,I6)
   53 FORMAT(/1X,'SIZE =',I4,'  INCREASES NSD =',I4//1X,' CORRECT ONLY
     1THESE CONSTRAINTS AT THIS CYCLE',I4)
 57   FORMAT(2X,6I4,2E15.5)
 58   FORMAT('  STRESS VIOLATIONS',2I5/3X,'SIZE IV  I6   M  MX  LC   XL'
     *)
  998 IF(SIZE.EQ.NSD) RETURN
  999 CONTINUE
      RETURN
      END
      SUBROUTINE ABSMAX(K,NN,INN,I6,NSD,IV,LDC,IPC,EP,NDAM)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/P4/INF( 50, 8),NGV( 14, 6),INO( 50),NDISP( 72)
      COMMON/R4/IIL( 50, 3),KLC( 50),IOK( 3),NO( 1)
      COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
```

```
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C***********************************************************************
C*    THIS SUBROUTINE -                                                *
C*       (1) CALCULATES MAX DISPL UNDER ALL NLC,
C*       (2) CHECKS DISPL CONSTRAINTS AND COMPUTE SENSITIVITY          *
C*           INFORMATION.                                              *
C***********************************************************************
   53 FORMAT(/1X,'SIZE =',I4,' INCREASES NSD =',I4//1X,' CORRECT ONLY
     1THESE CONSTRAINTS AT THIS CYCLE',I4)
  314 FORMAT(2X,5I4,2E15.5)
  315 FORMAT(3X,'DISPL. VIOLATIONS',I2/3X,'SIZE IV  I6  NC  LC      XL')
      WRITE(6,315) K
      DO 162 I=1,NN
      INN=INN+1
      DO 161 L=1,NLC
      T(4)=ZZ(I,L)
      T(5)= DZE(I)
      IF(T(4).LT.0.) T(5)=-T(5)
      T(6)=T(4)*T(5)
      IF(L.GT.1) GO TO 258
      YM=T(5)
      XL=T(6)
      LC=L
      GO TO 161
  258 IF(XL.GE.T(6)) GO TO 161
      YM=T(5)
      XL=T(6)
      LC=L
  161 CONTINUE
      IF((XL+EP).LT.1.) GO TO 162
      SIZE=SIZE+1
      IF(SIZE.GT.(NSD-NDAM-1))GO TO 259
      NDISP(INN)=SIZE
  212 DLPH(NDISP(INN))=1.0-XL
       DLP(NDISP(INN))=DABS(DLPH(NDISP(INN)))
      KLC(NDISP(INN))=LC
      IV=IV+1
      INO(IV)=NDISP(INN)
      INF(NDISP(INN),1)=I
      INF(NDISP(INN),2)=K
      INF(NDISP(INN),3)=0
      INF(NDISP(INN),4)=0
      INF(NDISP(INN),5)=LDC
      INF(NDISP(INN),6)=IV
      INF(NDISP(INN),7)=LC
      INF(NDISP(INN),8)=0
      ISIZ=NDISP(INN)
      NO(LC)=1
      IF(K.EQ.-1) GO TO 96
      I6=I6+1
      IIL(I6,K)=IV
      DO 158 J=1,N1
      L1=NZC( J,K)
  158 A2(L1,IV)=A2(L1,IV)+YM*Q(I,J,K)
      DS(I,I6)=YM
      GO TO 97
   96 A2(I,IV)=YM
   97 IF(IPC.NE.0) WRITE(6,314) ISIZ,IV,I6,I,LC,XL
  162 CONTINUE
```

```
      RETURN
  259 SIZE=SIZE-1
      WRITE(6,53) SIZE,NSU,NTL
      RETURN
      END
      SUBROUTINE GENC(NSD,NV,LX,IBU,IBUK,IV,IDC)                       SUB 13
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SV
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/V2/NIC( 3),IW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/P1/B1( 9, 9),B2( 9, 9),B3( 9, 9),ESF( 9, 9),NA( 156),NI1( 9
     1),NJ1( 9),NJ2( 9)
      COMMON/P2/XNUU( 14, 6),ELL(108, 2),BU( 14, 6),STRESS(1620),TCSM(
     1 156),TRCSSP(2808),XCOST( 3),ICSS( 108, 2),ISAC( 108, 2),INDC( 108
     2, 2),IGRT( 14, 6),IGRE( 108, 2),NNDC( 1080),LLN( 3),ITY( 3),ICSSM(
     3 108, 2)
      COMMON/P4/INF( 50, 8),NGV( 14, 6),INO( 50),NDISP( 72)
      COMMON/R2/PI(12, 1, 3),RR( 108, 2),E( 14, 6),MN( 108, 6),NOM( 14,
     1 6)
      COMMON/R4/IIL( 50, 3),KLC( 50),IOK( 3),NO( 1)
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A1/Q(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND( 216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
      COMMON/A7/DPX( 62, 50)
C***********************************************************************
C*    THIS SUBROUTINE - COMPUTES CAP LAMBDA.
C***********************************************************************
      INDEX=0
      IVV=IV+LX-1
      IF(IDC.GT.1) GO TO 77
      DO 75 I=LX,SIZE
      DO 75 J=1,NV
   75 DPX(J,I)=0.D0
      GO TO 76
   77 DO 78 II=1,IV
      I=INO(II)
      DO 78 J=1,NV
   78 DPX(J,I)=0.0
   76 DO 79 II=1,IV
      I=INO(II)
      IGR=NA(II)
      IF(IGR.EQ.0) GO TO 79
      DPX(IGR,I)=BE(II,1)
   79 CONTINUE
      LD=0
      DO 71 K=1,NSU
      CALL VARI(K)
      DO 71 III=1,3
      IF(ITY(III).EQ.0) GO TO 71
      MA=0
      LD=LD+1
      NGK=NG(LD)
      IF(NCI.EQ.0) GO TO 210
```

```
      DO 777 KK=1,NGK
      NJJ=NOM(KK,LD)
      MV=IGRT(KK,LD)
      IF(MV.EQ.0) GO TO 170
      DO 73 LC=1,NLC
      IF(NO(LC).EQ.0) GO TO 73
      DO 74 I=1,N1
   74  BE(I,LC)=0.D0
   73 CONTINUE
      DO 666 IM=1,NJJ
      IB3=0
      JP=0
      MA=MA+1
      MI=MN(MA,LD)
      MX=MP(MI,III)
      XX=BR(MI,III)
      IF(XX.EQ.0.0) GO TO 666
      XX=1.0
C
      CALL RECALL(III,LE,LS,LF,INDEX,MI,XX)
      DO 707 I=LS,LE
      II=NNDC(LF+I)
      IF(II.EQ.0) GO TO 707
      IF(II.GT.N1) GO TO 704
      IB3=IB3+1
       NA(IB3)=II
      JB3=0
      DO 703 J=LS,LE
      IJ=NNDC(LF+J)
      IF(IJ.EQ.0) GO TO 703
      IF(IJ.GT.N1) GO TO 703
      JB3=JB3+1
      B3(IB3,JB3)=ESF(I,J)
  703 CONTINUE
      GO TO 707
  704 II=II-N1
      JP=JP+1
      NI1(JP)=II
      JT=0
      JU=0
      DO 706 J=LS,LE
      IJ=NNDC(LF+J)
      IF(IJ.EQ.0) GO TO 706
      IF(IJ.GT.N1) GO TO 705
      JT=JT+1
      NJ2(JT)=IJ
      B2(JP,JT)=ESF(I,J)
      GO TO 706
  705 IJ=IJ-N1
      JU=JU+1
      NJ1(JU)=IJ
      B1(JP,JU)=ESF(I,J)
  706 CONTINUE
  707 CONTINUE
      DO 124 LC=1,NLC
      IF(NO(LC).EQ.0) GO TO 124
      IF(IB3.EQ.0) GO TO 80
      DO 123 J=1,IB3
      J1=NA(J)
      L1=NZC(J1,K)
      DO 123 I=1,IB3
```

```
      J2=NA(I)
  123 BE(J2,LC)=BE(J2,LC)-B3(I,J)*ZB(L1,LC)
   80 IF(MX.EQ.-1) GO TO 124
      DO 518 J=1,JP
  518 Z(J,LC)=0.D0
      IF(JT.EQ.0) GO TO 511
      DO 118 J=1,JP
      J1=NI1(J)
      DO 118 L=1,JT
      J2=NJ2(L)
      L1=NZC(J2,K)
  118 Z(J,LC)=Z(J,LC)-B2(J,L)*ZB(L1,LC)
      DO 122 I=1,JT
      J2=NJ2(I)
      DO 122 J=1,JP
      J1=NI1(J)
  122 BE(J2,LC)=BE(J2,LC)-B2(J,I)*ZI(J1,LC,K)
  511 DO 116 J=1,JP
      DO 116 L=1,JU
      J2=NJ1(L)
  116 Z(J,LC)=Z(J,LC)-B1(J,L)*ZI(J2,LC,K)
      DO 125 I=1,N1
      DO 125 J=1,JP
      J1=NI1(J)
  125 BE( I,LC)=BE( I,LC)+Q(J1,I,K)*Z(J,LC)
  124 CONTINUE
      I6=IOK(K)
      IF(I6.EQ.0.OR.MX.EQ.-1) GO TO 666
      DO 514 I=1,I6
      IZZ=IIL(I,K)
      IZ=INO(IZZ)
      LC=KLC(IZ)
      DO 514 J=1,JP
      J1=NI1(J)
  514 DPX(MV,IZ)=DPX(MV,IZ)+Z(J,LC)*DPZ(J1,IZZ)
  666 CONTINUE
      DO 117 II=1,IV
      I=INO(II)
      LC=KLC(I)
      DO 117 J=1,N1
      L1=NZC(J,K)
  117 DPX(MV,I)= DPX(MV,I)+ BE(J,LC)*DS(L1,II)
      GO TO 777
  170 MA=MA+NJJ
  777 CONTINUE
      GO TO 71
C     WHEN NCI IS ZERO.
  210 DO 889 KK=1,NGK
      NJJ=NOM(KK,LD)
      MV=IGRT(KK,LD)
      IF(MV.EQ.0) GO TO 720
      DO 888 IM=1,NJJ
      L=0
      MA=MA+1
      MI=MN(MA,LD)
      XX=BR(MI,III)
      IF(XX.EQ.0.0) GO TO 888
      XX=1.0
      CALL RECALL(III,LE,LS,LF,INDEX,MI,XX)
      DO 714 I=LS,LE
      II=NNDC(LF+I)
```

```
      IF(II.EQ.0) GO TO 714
      L=L+1
      NA(L)=II
      M=0
      DO 712 J=LS,LE
      IJ=NNDC(LF+J)
      IF(IJ.EQ.0) GO TO 712
      M=M+1
      BI(L,M)=ESF(I,J)
  712 CONTINUE
  714 CONTINUE
      DO 200 LC=1,NLC
      IF(NO(LC).EQ.0) GO TO 200
      DO 156 I=1,L
      BE(I,LC)=0.D0
      DO 156 J=1,L
      J2=NA(J)
      L2=NZC(J2,K)
  156 BE(I,LC)=BE(I,LC)-BI(I,J)*ZB(L2,LC)
  200 CONTINUE
      DO 158 II=1,IV
      I=INO(II)
      LC=KLC(I)
      DO 158 J=1,L
      J2=NA(J)
      L2=NZC(J2,K)
  158 DPX(MV,I)= DPX(MV,I)+ BE(J,LC)*DS(L2,II)
  888 CONTINUE
      GO TO 889
  720 MA=MA+NJJ
  889 CONTINUE
   71 CONTINUE
C     DO 1000 J=LX,SIZE
C     WRITE(6,1001) J
C1000 WRITE(6, 10)  (DPX(I,J),I=1,NV)
C  10 FORMAT(3X,' CAP LAMBDA*TRANSPOSE'/(3X,10E12.4))
C1001 FORMAT(3X,'SIZE=',I3)
      RETURN
      END
      SUBROUTINE DELBE(IJ,NDC,NV,*)                                SUB 14
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C*****************************************************************
C*    THIS SUBROUTINE COMPUTES DELTA B VECTOR, I.E. CHANGES IN DESIGN  *
C*    VARIABLES.  LAGRANGE MULTIPLIERS ARE COMPUTED AND THEIR SIGNS    *
C*    ARE CHECKED.  CONSTRAINTS CORRESPONDING TO NEGATIVE MULTIPLIERS  *
C*    ARE TAKEN OUT OF THE VIOLATED CONSTRAINT SET                     *
C*    IJ    -  NO. OF STRESS & DISPLACEMENT VIOLATIONS                 *
C*    NDC    -  NO. OF DESIGN VARIABLE CONSTRAINT VIOLATIONS           *
C*    SIZE   -  TOTAL NO. OF CONSTRIANT VIOLATIONS                     *
C*    Z      -  DELTA B VECTOR ON RETURN                               *
C*****************************************************************
   26 FORMAT( /1X,'REQUESTED CHANGES IN CONSTRAINTS DEL PHI'/(4(I5,E12.4
```

```
     1)))
  40 FORMAT( /1X,'LAGRANGE MULTIPLIERS'/(4(I5,E12.4)))
      IF(IJ .GT. 0 ) GO TO 448
      CALL DESVV(IJ,NDC,NV)
      RETURN
 448 YM=STEP
     IF(YM.GT.0.) GO TO 466
     YM=1.
     DO 467 I=1,NV
 467 Z(I,1)=0.D0
C.... COMPUTE RIGHT HANDSIDE SIDE OF THE LAGRANGE MULTIPLIER EQUATIONS
 466 DO 468 I=1,NV
 468 T(I)=Z(I,1)*RO(I)
     DO 225 I=1,IJ
     ZZ(I,2)=-DLPH(I)
     ZZ(I,1)=0.
     DO 225 J=1,NV
 225 ZZ(I,1)=ZZ(I,1)-DPB(J,I)*T(J)
     IF(IJ.EQ.SIZE) GO TO 159
     DO 420 I=1,NDC
     K=H(I)
     J=IJ+I
     ZZ(J,2)=-DLPH(J)
 420 ZZ(J,1)=-DZE(I)*Z(K,1)*WM(K)
 159 CONTINUE
     WRITE(6,26) (I,DLPH(I),I=1,SIZE)
C.... COMPUTE (CAP LAMDA TRANSPOSE )*(CAP LAMBDA ),(SIZE,SIZE)
     DO 166 I=1,IJ
     DO 166 J=I,IJ
     DPZ(I,J)=0.
     DO 161 K=1,NV
 161 DPZ(I,J)=DPZ(I,J)+DPB(K,I)*DPB(K,J)
 166 DPZ(J,I)=DPZ(I,J)
C.... COMPUTE LAGRANGE MULTIPLIERS
     IF(IJ.EQ.SIZE) GO TO 421
     CALL SOD(IJ,NDC,YM,NV,&205,&159)
      RETURN
 421 CONTINUE
     CALL SOLVEL(SIZE,ER,5)
     DO 424 I=1,SIZE
 424 T(I)=BE(I,1)+BE(I,2)/YM
     WRITE(6,40)  (I,T(I),I=1,SIZE)
C... CHECK SIGN OF LAGRANGE MULTIPLIERS
      N=0
     DO 235 I=1,SIZE
     IF(T(I).LE.0.) GO TO 235
     N=N+1
     ND(N)=I
 235 CONTINUE
     IF(N.EQ.SIZE) GO TO 250
     SIZE=N
     IJ=N
      IF(N.EQ.0) RETURN 1
     DO 240 I=1,SIZE
     IF(I.EQ.ND(I)) GO TO 240
     DLPH(I)=DLPH(ND(I))
     DLP(I)=DLP(ND(I))
     ZZ(I,1)=ZZ(ND(I),1)
     ZZ(I,2)=ZZ(ND(I),2)
     DO 241 J=1,NV
 241 DPB(J,I)=DPB(J,ND(I))
```

```
  240 CONTINUE
      GO TO 159
  250 CONTINUE
      DO 910 I=1,SIZE
      ZZ(I,1)=BE(I,1)
  910 ZZ(I,2)=BE(I,2)
      DO 206 I=1,NV
      BE(I,1)=-Z(I,1)*RO(I)
      BE(I,2)=0.
      DO 912 J=1,SIZE
      BE(I,1)=BE(I,1)-DPB(I,J)*ZZ(J,1)
  912 BE(I,2)=BE(I,2)-DPB(I,J)*ZZ(J,2)
      BE(I,1)=BE(I,1)*RO(I)
  206 BE(I,2)=BE(I,2)*RO(I)
       RETURN
  205 RETURN 1
      END
       SUBROUTINE  DESVV(IJ,NDC,NV)                                    SUB  15
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C***********************************************************************
C*    THIS SUBROUTINE COMPUTES DELTA B VECTOR WHEN ONLY DESIGN VARIABLE*
C*    CONSTRAINTS ARE VIOLATED                                         *
C***********************************************************************
      DO 449 I=1,NV
      BE(I,1)=0.
      BE(I,2)=0.
  449 ND(I)=I
      DO 446 I=1,NDC
      K=H(I)
      ND(K)=0
      W(K)=DLPH(IJ+I)/DZE(I)
      BE(K,2)=W(K)
  446 CONTINUE
      DO 451 I=1,NV
      IF(ND(I).EQ.0) GO TO 451
      W(I)=-STEP*Z(I,1)
      BE(I,1)=-Z(I,1)
  451 CONTINUE
       RETURN
      END
      SUBROUTINE SDD(IJ,NDC,YM,NV,*,*)                                 SUB  16
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A4/X( 108, 3),DLP( 60),DLPH( 60),T( 156),WM( 51),RO( 51)
      COMMON/A5/D( 36, 24),DS( 36, 50),A2( 36, 50),DKI(12,36),KIIUBW( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C***********************************************************************
C*    THIS SUBROUTINE COMPUTES DELTA B VECTOR WHEN STRESS & DISPLACE-  *
```

```
C*     MENT & DESIGN VARIABLE CONSTRAINTS ARE VIOLATED                      *
C***************************************************************************
   40 FORMAT( /1X,'LAGRANGE MULTIPLIERS'/(4(I5,E12.4)))
      DO 422 J=1,NDC
      K=H(J)
        Z(J,2)=1.0D0/(DZE(J)*WM(K)*DZE(J))
      DO 422 I=1,IJ
  422  DS(J,I)=DZE(J)*DPB(K,I)*RO(K)
      DO 423 I=1,IJ
      DO 423 J=I,IJ
      DO 447 K=1,NDC
  447 DPZ(I,J)=DPZ(I,J)- DS(K,I)* DS(K,J)*Z(K,2)
  423 DPZ(J,I)=DPZ(I,J)
      DO 200 I=1,SIZE
      VV(I)=ZZ(I,1)
  200 W(I)=ZZ(I,2)
      DO 425 I=1,IJ
      DO 425 J=1,NDC
      ZZ(I,1)=ZZ(I,1)- DS(J,I)*ZZ(IJ+J,1)*Z(J,2)
  425 ZZ(I,2)=ZZ(I,2)- DS(J,I)*ZZ(IJ+J,2)*Z(J,2)
      CALL SOLVEL(IJ,ERR5)
      DO 431 I=1,NDC
      K=I+IJ
      BE(K,1)=VV(K)
      BE(K,2)=W(K)
      DO 460 J=1,IJ
      BE(K,1)=BE(K,1)- DS(I,J)*BE(J,1)
  460 BE(K,2)=BE(K,2)- DS(I,J)*BE(J,2)
      BE(K,1)=BE(K,1)*Z(I,2)
  431 BE(K,2)=BE(K,2)*Z(I,2)
      DO 201 I=1,SIZE
  201 T(I)=BE(I,1)+BE(I,2)/YM
      WRITE(6,40) (I,T(I),I=1,SIZE)
C.....CHECK SIGN OF LAGRANGE MULTIPLIERS
      N=0
      DO 432 I=1,IJ
C
      IF(T(I).LE.0.) GO TO 432
      N=N+1
      ND(N)=I
  432 CONTINUE
      IF(N.GT.0) GO TO 445
      CALL DESVV(IJ,NDC,NV)
       RETURN
  445 JI=IJ
      IF(N.EQ.IJ) GO TO 433
      IJ=N
      DO 434 I=1,IJ
      IF(I.EQ.ND(I)) GO TO 434
      DO 435 J=1,NV
  435 DPB(J,I)=DPB(J,ND(I))
  434 CONTINUE
  433 DO 436 I=1,NDC
      K=JI+I
      IF(T(K).LT.0.) GO TO 436
      N=N+1
      ND(N)=K
  436 CONTINUE
      IF(N.EQ.SIZE) GO TO 437
      SIZE=N
       IF(N.EQ. 0) RETURN
```

```
      M=NDC
      NDC=N-IJ
      DO 438 I=1,SIZE
      DLPH(I)=DLPH(ND(I))
      DLP(I)=DLP(ND(I))
      ZZ(I,1)=VV(ND(I))
      ZZ(I,2)=W(ND(I))
  438 CONTINUE
       IF(NDC.EQ.M.OR.NDC.EQ.0) RETURN 2
      DO 439 I=1,NDC
      J=ND(I+IJ)-JI
      H(I)=H(J)
  439 DZE(I)=DZE(J)
      RETURN 2
  437 CONTINUE
      DO 440 I=1,SIZE
      ZZ(I,1)=BE(I,1)
      ZZ(I,2)=BE(I,2)
  440 CONTINUE
      DO 450 I=1,NV
      BE(I,1)=0.
  450 BE(I,2)=0.
      DO 441 I=1,NDC
      K=H(I)
      L=IJ+I
      BE(K,1)=DZE(I)*ZZ(L,1)* WM(K)
  441 BE(K,2)=-DZE(I)*ZZ(L,2)*WM(K)
      DO 442 I=1,NV
      SM1=0.0D0
      SM2=0.0D0
      DO 443 J=1,IJ
      SM1=SM1+DPB(I,J)*ZZ(J,1)
  443 SM2=SM2-DPB(I,J)*ZZ(J,2)
      BE(I,1)=-BE(I,1)-(SM1+Z(I,1)*RO(I))*RO(I)
  442 BE(I,2)=BE(I,2)+SM2*RO(I)
       RETURN
      END
      SUBROUTINE SOLVEL(NF,ER)                                          SUB  17
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/A3/BR( 108, 2),TRSF( 108, 1),CSTF(48, 1, 4),SSPF( 1, 1, 3),
     1Z( 51, 3),DZE( 60),MP( 108, 2),ND(  216)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),W( 72),H( 108),VV(
     1 156),Y( 108, 3),NZC( 24, 3)
C*****************************************************************
C*    GAUSSIAN ELLIMINATION PROCESS                                 *
C*    TOTAL PIVOTING IS USED                                        *
C*    DPZ IS THE SQUARE MATRIX(LHS OF EQ.)
C*    MATRIX ZZ IS THE RHS 06 5QUIT -N2                             *
C*    ZZ IS SAVED                                                   *
C*    FINAL SOLUTION IS IN MATRIX BE                                *
C*****************************************************************
      ER=0.000001
      M=2
      IF(NF.GT.1) GO TO 76
      IF(DPZ(1,1).EQ.0.) GO TO 76
      A=1./DPZ(1,1)
      DO 77 J=1,M
   77 BE(1,J)=ZZ(1,J)*A
      GO TO 999
   76 WRITE(6,41) NF
      DO 78 J=1,M
```

```
   78 BE(I,J)=0.
      GO TO 999
   75 NMP=NF-1
      DO 10 I=1,NF
      ND(I)=I
      DO 10 J=1,M
   10  BE(I,J)=ZZ(I,J)
      DO 400 K=1,NMP
C**** SEARCH FOR THE PIVOT ELEMENT
      IB=0
      JB=0
      A=0.
      DO 20 I=K,NF
      DO 20 J=K,NF
      IF(DABS(DPZ(I,J))-A) 20,20,31
   31 A=DABS(DPZ(I,J))
      IB=I
      JB=J
   20 CONTINUE
      IF(A-ER) 40,40,42
   40 WRITE(6,41) K,IB,JB
   41 FORMAT(1X, 'WHOOPS DEPENDENT EQUATIONS',3I4)
      DO 43 I=K,NF
      DO 44 J=K,NF
      DPZ(I,J)=0.
      IF(I.EQ.J) DPZ(I,J)=1.0
   44 CONTINUE
      DO 43 J=1,M
   43  BE(I,J)=0.
      GO TO 800
C**** INTERCHANGE ROWS AND COLUMNS
   42 IF(IB-K) 51,51,50
   50 DO 60 J=K,NF
      A=DPZ(K,J)
      DPZ(K,J)=DPZ(IB,J)
   60 DPZ(IB,J)=A
      DO 63 J=1,M
      A= BE(K,J)
       BE(K,J)= BE(IB,J)
   63  BE(IB,J)=A
   51 IF(JB-K) 62,62,61
   61 DO 70 I=1,NF
      A=DPZ(I,K)
      DPZ(I,K)=DPZ(I,JB)
   70 DPZ(I,JB)=A
C**** KEEP TRACK OF COLUMNS
      J=ND(K)
      ND(K)=ND(JB)
      ND(JB)=J
   62 A=DPZ(K,K)
      KP=K+1
C**** DIVIDE THE PIVOT ROW BY THE PIVOT ELEMENT
      DO 80 J=K,NF
   80 DPZ(K,J)=DPZ(K,J)/A
      DO 81 J=1,M
   81  BE(K,J)= BE(K,J)/A
C**** PERFORM ELLIMINATION
      DO 82 I=KP,NF
      A=DPZ(I,K)
      DO 83 J=1,M
   83  BE(I,J)= BE(I,J)-A*BE(K,J)
```

```
      DO 82 J=K,NF
      DPZ(I,J)=DPZ(I,J)-A*DPZ(K,J)
   82 CONTINUE
  400 CONTINUE
  800 CONTINUE
      IF(DABS(DPZ(NF,NF)).GT.ER) GO TO 500
      DPZ(NF,NF)=1.
      DO 501 J=1,M
  501 BE(NF,J)=0.0
  500 CONTINUE
      DO 90 J=1,M
   90  BE(NF,J)= BE(NF,J)/DPZ(NF,NF)
      DO 91 L=1,M
      DO 100 KK=1,NMP
      K=NF-KK
      KP=K+1
      DO 100 J=KP,NF
  100  BE(K,L)= BE(K,L)-DPZ(K,J)*BE(J,L)
   91 CONTINUE
C**** REARRANGE THE SOLUTION MATRIX
      DO 111 I=1,NF
      DO 111 J=1,M
  111 DPZ(I,J)=BE(I,J)
      DO 110 I=1,NF
      DO 110 J=1,M
  110 BE(ND(I),J)=DPZ(I,J)
  999 RETURN
      END
      SUBROUTINE SUBSP(N,MK,ITMAX,ERR,IDC,IIX8)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER SIZE,BNC,SN
      COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU
      COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N11,ISQ,IQ1
      COMMON/V2/NIC( 3),NW( 6),NG( 6),NBW1( 3),NBW2( 3),NBW3( 3),NM( 6),
     1NBJ( 3),NJ( 3),NCB( 3),NEW( 3),IQS( 3),MEB( 6),MEF( 6)
      COMMON/R5/B( 14, 6),SL( 14, 6),SU( 14, 6),DPB( 51, 50),DLIM(12, 3)
     1,SS( 51)
      COMMON/A1/G(12, 24, 3),ZI(12, 1, 3),C( 36, 24),ZB( 36, 1)
      COMMON/A5/XCL( 36, 24),AMAS2( 36, 50),XM( 36, 50),DKI(12,36),KIIUB
     1W( 3)
      COMMON/A6/DPZ( 50, 50),ZZ( 72, 3),BE( 108, 3),F( 72),H( 108),VV(
     1 156),A( 108, 3),NZC( 24, 3)
      COMMON/C1/X( 72, 2),Y( 72, 2),W( 2),DM(  1,  1),IETA( 7)
     $/C3/ QQK( 2, 2),QOM( 2, 2),Q( 2, 2)
      MM=MK
      IP=1
      WL=1.D 25
      ITER=0
      IEND=0
      RELERR=0.05
      IQ=MINO(IP*2,IP+8,N)
      GO TO 30
    5 ITER=ITER+1
C     SOLVING X-BAR
C
C     COMPUTE THE EFFECTIVE BOUNDARY EIGEN VECTOR
C
      DO 121 I=1,BNC
      DO 121 J=1,IQ
  121 AMAS2(I,J)=0.D0
      NCX=BNC
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      DO 101 K=1,NSU
      CALL VARI(K)
      IF(NCI.EQ.0)GO TO 101
      DO 102 I=1,NI
      L1=NZC(I,K)
      DO 102 L=1,IQ
      QTYI=0.D0
      DO 202 J=1,NCI
      I6=NCX+J
  202 QTYI=QTYI+G(J,I,K)*Y(I6,L)
      AMAS2(L1,L)=AMAS2(L1,L)+QTYI
  102 CONTINUE
      NCX=NCX+NCI
  101 CONTINUE
      DO 122 I=1,BNC
      DO 122 J=1,IQ
  122 AMAS2(I,J)=AMAS2(I,J)+Y(I,J)
      IF(NCI.EQ.0)GO TO 124
      DO 123 I=1,BNC
      DO 123 J=1,NBW
  123 XCL(I,J)=C(I,J)
C
C
  124 CONTINUE
      CALL SOLDUP(IQ,NBW,BNC)
      DO 111 I=1,BNC
      DO 111 L=1,IQ
  111 X(I,L)=AMAS2(I,L)
C
C
C
      NCX=BNC
      DO 108 K=1,NSU
      CALL VARI(K)
      IF(NCI.EQ.0)GO TO 108
      IQQ=KIIUBW(K)
      DO 103 J=1,NU3
      IQQ=IQQ+1
      DO 103 I=1,NCI
      IF(IDC.GT.1)GO TO 104
      XCL(I,J)=DKI(I,IQQ)
      GO TO 103
  104 XCL(I,J)=DPB(I,IQQ)
  103 CONTINUE
      DO 105 I=1,NCI
      I6=NCX+I
      DO 105 L=1,IQ
  105 AMAS2(I,L)=Y(I6,L)
C
      CALL SOLDUP(IQ,NU3,NCI)
C
      DO 113 I=1,NCI
      DO 114 LL=1,IQ
      DO 109 L=1,NI
      L1=NZC(L,K)
      QXB=G(I,L,K)*X(L1,LL)
C
      AMAS2(I,LL)=AMAS2(I,LL)+QXB
C
  109 CONTINUE
  114 CONTINUE
```

```
 113  CONTINUE
C
C
C
      DO 116 I=1,NCI
      I6=NCX+I
      DO 116 L=1,IQ
      X(I6,L)=AMAS2(I,L)
 116  CONTINUE
      NCX=NCX+NCI
 108  CONTINUE
C   PROJECTED STIFFNESS MATRIX QQK
      DO 21 I=1,IQ
      DO 21 J=I,IQ
      S=0.D00
      DO 22 K=1,N
   22 S=S+X(K,I)*Y(K,J)
      QQK(I,J)=S
   21 QQK(J,I)=S
C   INTERMEDIATE VECTORS Y FOR ITER=0, AND Y-BAR FOR ITER O
 30      NN=SN/2
      I8=IIX8
      CALL MEVEC(NN,N,IDC,I8,2)
      IF(ITER) 5,5,40
C   PROJECTED MASS MATRIX
   40 DO 41 I=1,IQ
      DO 41 J=I,IQ
      S=0.D 00
      DO 42 K=1,N
   42 S=S+X(K,I)*Y(K,J)
      QQM(I,J)=S
   41 QQM(J,I)=S
      IF(RELERR.GT.0.1) RELERR=0.1
      THRESH=0.1*RELERR
C   SUBSPACE EIGENVALUES W AND EIGENMATRIX Q
   50 CALL JACOBI(IQ,ITMAX,THRESH)
C   SORTING EIGENVALUES IN INCREASING ORDER
      IF(MOD(ITER-1,11))60,80,60
C   RELATIVE ERROR CHECK
   60 WLT=W(IP)
      RELERR=DABS(1.-WL/WLT)
      IF(ITER.GT.ITMAX) GO TO 65
      IF(RELERR-ERR) 65,65,70
C   GETTING EIGENVECTORS IN ORIGINAL SPACE
   65 IEND=1
      DO 66 I=1,N
      DO 66 J=1,IQ
   66 Y(I,J)=X(I,J)
C   TRANSFORMING INTERMEDIATE VECTORS
   70 DO 71 I=1,N
      DO 71 J=1,IQ
      S=0.D 00
      DO 72 K=1,IQ
   72 S=S+Y(I,K)*Q(K,J)
   71 X(I,J)=S
      IF(IEND) 75,75,80
   75 DO 73 I=1,N
      DO 73 J=1,IQ
   73 Y(I,J)=X(I,J)
      WL=WLT
      GO TO 5
```

```
C     SORTING ROUTINE
   80 IQM=IQ-1
      DO 81 II=1,IQM
      WMIN=W(II)
      IMIN=II
      III=II+1
      DO 82 I=III,IQ
      IF(WMIN.LT.W(I)) GO TO 82
      WMIN=W(I)
      IMIN=I
   82 CONTINUE
      IF(IMIN.EQ.II) GO TO 81
      S=W(II)
      W(II)=W(IMIN)
      W(IMIN)=S
      DO 83 J=1,N
      S=X(J,II)
      X(J,II)=X(J,IMIN)
      X(J,IMIN)=S
   83 CONTINUE
   81 CONTINUE
      IF(IEND) 60,60,90
C     NORMALIZING EIGENVECTORS
   90 DO 91 J=1,IQ
      S=0.D 00
      DO 92 I=1,N
   92 S=S+X(I,J)*X(I,J)
      S=1.D 00/DSQRT(S)
      DO 93 I=1,N
   93 X(I,J)=X(I,J)*S
   91 CONTINUE
C     PRINT OUT FOR INFORMATIONS
      WRITE(6,6000) ITER,RELERR,(W(I),I=1,IQ)
 6000 FORMAT(/' ITER=',I5,5X,'RELERR=',E13.5/' EIGENVALUES',(5E15.6))
      IIX8=I8
      RETURN
      END
      SUBROUTINE JACOBI(N,ITMAX,THRESH)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/A5/XCL( 36, 24),AMAS2( 36, 50),ZM( 36, 50),DKI(12,36),KIIUB
     1W( 3)
      COMMON/C1/Z( 72, 2),Y( 72, 2),W( 2),DM(  1,  1),IETA( 7)
     $/C3/ XK( 2, 2),XM( 2, 2),P( 2, 2)
C     SOLVE XK * P = XM * P * DIAG(W) FOR ALL EIGENVALUES AND VECTORS
      DO 1 I=1,N
      DO 2 J=1,N
    2 P(I,J)=0.
    1 P(I,I)=1.
      NM1=N-1
      ISMAL=0
      ITER=0
  100 CFMX=0.
      DO 10 I=1,NM1
      XMI=XM(I,I)
      XKI=XK(I,I)
      IP1=I+1
      DO 10 J=IP1,N
      XMJ=XM(J,J)
      XKJ=XK(J,J)
      XMIJ=XM(I,J)
      XKIJ=XK(I,J)
```

```
      CFM=XMIJ*XMIJ/XMI/XMJ
      CFK=XKIJ*XKIJ/XKI/XKJ
      CF=DMAX1(CFM,CFK)
      IF(CFMX.LT.CF) CFMX=CF
      IF(CF.LT.0.1*THRESH) GO TO 10
      BKI=XKI*XMIJ-XMI*XKIJ
      BKJ=XKJ*XMIJ-XMJ*XKIJ
      BK2=(XKI*XMJ-XKJ*XMI)*0.5
      SX=BK2*BK2+BKI*BKJ
      IF(SX.LT.0.) SX=0.
      X=BK2+DSIGN(DSQRT(SX),BK2)
      GAM=-BKI/X
      ALP=BKJ/X
      DO 20 L=1,N
      TK=XK(L,I)
      TM=XM(L,I)
      TP=P(L,I)
      XK(L,I)=TK+XK(L,J)*GAM
      XK(L,J)=TK*ALP+XK(L,J)
      XM(L,I)=TM+XM(L,J)*GAM
      XM(L,J)=TM*ALP+XM(L,J)
       P(L,I)=TP+ P(L,J)*GAM
       P(L,J)=TP*ALP+ P(L,J)
   20 CONTINUE
      DO 21 L=1,N
      TK=XK(I,L)
      TM=XM(I,L)
      XK(I,L)=TK+GAM*XK(J,L)
      XK(J,L)=XK(J,L)+ALP*TK
      XM(I,L)=TM+GAM*XM(J,L)
      XM(J,L)=XM(J,L)+ALP*TM
   21 CONTINUE
      ISMAL=ISMAL+1
   10 CONTINUE
      ITER=ITER+1
      IF(CFMX.LT.THRESH) GO TO 44
      IF(ITER.LT.ITMAX) GO TO 100
   44 DO 30 L=1,N
      W(L)=XK(L,L)/XM(L,L)
      XMJ=DABS(XM(L,L))
      T=1./DSQRT(XMJ)
      DO 30 M=1,N
      P(M,L)=P(M,L)*T
   30 CONTINUE
      RETURN
      END
//GO.SYSIN DD *
```

Example: Closed Tail-Boom with 6 Damage Conditions

```
    0     3     3     5     1    51    72    36    24    60    50     0
    2     1     0     1     1     1     0     0     1     1     1
    3     0     3    10     2     6     9     0     0
   0.0000     0.0020     0.2500     1.0000     0.0010     1.0000     1.0000
   1.0000     1.0000     1.0000    29.0000    -1.0000
  0.1000000E-05   0.1000000E-03   0.1000000E-03   0.1000000E-03   0.1000000E-05
   0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
   0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
   0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
   0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
   0.5        0.5        0.5        0.5
   4
   1     2     3     4
   1     1.4903     1.6918     0.0
```

```
 2     1.4903    -1.3658     0.0
 3    -1.4903     1.6918
 4    -1.4903    -1.3658
 0
12     4    12    12    21    12    12
 9    10    11    12
 1    66.500    -9.401      9.855     1     1     1
 2    66.500    -9.239     -9.855     1     1     1
 3    66.500     9.401      9.855     1     1     1
 4    66.500     9.239     -9.855     1     1     1
 5    33.500   -10.666     11.105     1     1     1
 6    33.500   -10.485    -11.105     1     1     1
 7    33.500    10.666     11.105     1     1     1
 8    33.500    10.485    -11.105     1     1     1
 9     0.000   -11.950     12.375
10     0.000   -11.750    -12.375
11     0.000    11.950     12.375
12     0.000    11.750    -12.375
 0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
 0.5        0.5        0.5        0.5
 0
36    14    14     1    36
 1     2     2     3
 2     2     1     4
 3     4     5     6     9    10
 4     4     7     8    11    12
 5     2    13    15
 6     2    14    16
 7     2    17    18
 8     2    20    21
 9     2    19    22
10     4    23    24    27    28
11     4    25    26    29    30
12     2    31    33
13     2    32    34
14     2    35    36
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 0.0415  100.0000    1.0000   25.0000   25.0000    0.1000    0.3000  10500.00
 1     7     3     0     0
 2     5     1     0     0
 3     6     2     0     0
 4     8     4     0     0
 5     7     1     0     0
 6     5     3     0     0
 7     5     2     0     0
 8     6     1     0     0
 9     6     4     0     0
10     8     2     0     0
11     7     4     0     0
```

```
12     8     3     0     0
13     3     1     0    -1
14     1     2     0    -1
15     2     4     0    -1
16     4     3     0    -1
17     3     2     0    -1
18     4     1     0    -1
19    11     7     0    +1
20     9     5     0    +1
21    10     6     0    +1
22    12     8     0    +1
23    11     5     0    +1
24     9     7     0    +1
25     9     6     0    +1
26    10     5     0    +1
27    10     8     0    +1
28    12     6     0    +1
29    11     8     0    +1
30    12     7     0    +1
31     7     5     0    +1
32     5     6     0    +1
33     6     8     0    +1
34     8     7     0    +1
35     7     6     0    +1
36     8     5     0    +1
16     3     3     1    16
15     4     1     2     3     4
16     4     5     6     7     8
17     8     9    10    11    12    13    14    15    16
  0.0200      0.0500      1.0000     40.2000     40.2000        0.0980      0.3000   10400.00
  0.0200      0.0500      1.0000     40.2000     40.2000        0.0980      0.3000   10400.00
  0.0200      0.0500      1.0000     40.2000     40.2000        0.0980      0.3000   10400.00
 1     7     9    11     1
 2     7     9     5     1
 3     5     3     7
 4     5     3     1
 5     6    12    10     1
 6     6    12     8     1
 7     2     8     5
 8     2     8     4
 9     6     9    10     1
10     6     9     5     1
11     2     5     6
12     2     5     1
13    12     7    11     1
14    12     7     8     1
15     8     3     7
16     8     3     4
12     8    24    12    33    12    12
 5     6     7     8     9    10    11    12
 1   127.500      -7.063       7.543     1     1     1
 2   127.500      -6.937      -7.543     1     1     1
 3   127.500       7.063       7.543     1     1     1
 4   127.500       6.937      -7.543     1     1     1
 5    66.500      -9.401       9.855     1     1     1
 6    66.500      -9.239      -9.855     1     1     1
 7    66.500       9.401       9.855     1     1     1
 8    66.500       9.239      -9.855     1     1     1
 9    99.500      -8.136       8.604     1     1     1
10    99.500      -7.994      -8.604     1     1     1
11    99.500       8.136       8.604     1     1     1
```

```
12    99.500     7.994     -3.604     1     1     1
 0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
 0.5        0.5        0.5        0.5
 4
 9    10    11    12
 9     0.0          0.0          -0.140
10     0.0          0.0          -0.140
11     0.0          0.0          -0.140
12     0.0          0.0          -0.140
36    14    14    37    72
18     2    38    39
19     2    37    40
20     4    41    42    45    46
21     4    43    44    47    48
22     2    49    51
23     2    50    52
24     2    53    54
25     2    56    57
26     2    55    58
27     4    59    60    63    64
28     4    61    62    65    66
29     2    67    69
30     2    68    70
31     2    71    72
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
 0.0415   100.0000    1.0000    25.0000    25.0000    0.1000    0.3000   10500.00
37    11     3     0
38     9     1
39    10     2
40    12     4
41    11     1
42     9     3
43     9     2
44    10     1
45    10     4
46    12     2
47    11     4
48    12     3
49     3     1     0    -1
50     1     2     0    -1
51     2     4     0    -1
52     4     3     0    -1
53     3     2     0    -1
54     4     1     0    -1
55     7    11
56     5     9
57     6    10
58     8    12
59     7     9
```

```
60      5     11
61      5     10
62      6      9
63      6     12
64      8     10
65      7     12
66      8     11
67     11      9      0     +1
68      9     10      0     +1
69     10     12      0     +1
70     12     11      0     +1
71     11     10      0     +1
72     12      9      0     +1
16      3      3     17     32
32      4     17     18     19     20
33      4     21     22     23     24
34      8     25     26     27     28     29     30     31     32
  0.0200      0.0500      1.0000    40.2000    40.2000     0.0980     0.3000   10400.00
  0.0200      0.0500      1.0000    40.2000    40.2000     0.0980     0.3000   10400.00
  0.0200      0.0500      1.0000    40.2000    40.2000     0.0980     0.3000   10400.00
17      5     11      7
18      5     11      9
19      9      3     11
20      9      3      1
21     10      8      6
22     10      8     12
23     12      2      4
24     12      2     10
25     10      5      6
26     10      5      9
27      2      9     10
28      2      9      1
29      8     11      7
30      8     11     12
31     12      3     11
32     12      3      4
12      8     24     12     33     12     12
 1      2      3      4      5      6      7      8
 1   173.500     -5.300      5.800     1     1     1
 2   173.500     -5.200     -5.800     1     1     1
 3   173.500      5.300      5.800     1     1     1
 4   173.500      5.200     -5.800     1     1     1
 5   127.500     -7.063      7.543     1     1     1
 6   127.500     -6.937     -7.543     1     1     1
 7   127.500      7.063      7.543     1     1     1
 8   127.500      6.937     -7.543     1     1     1
 9   151.500     -6.143      6.634     1     1     1
10   151.500     -6.031     -6.634     1     1     1
11   151.500      6.143      6.634     1     1     1
12   151.500      6.031     -6.634     1     1     1
  0.5        0.5        0.5        0.5        0.5        0.5        0.5        0.5
  0.5        0.5        0.5        0.5
0
36     14     14     73    108
35      2     74     75
36      2     73     76
37      4     77     78     81     82
38      4     79     80     83     84
39      2     85     87
40      2     86     88
41      2     89     90
```

42	2	92	93						
43	2	91	94						
44	4	95	96	99	100				
45	4	97	98	101	102				
46	2	103	105						
47	2	104	106						
48	2	107	108						
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
0.0415	100.0000	1.0000	25.0000	25.0000	0.1000	0.3000	10500.00		
73	11	3							
74	9	1							
75	10	2							
76	12	4							
77	11	1							
78	9	3							
79	9	2							
80	10	1							
81	10	4							
82	12	2							
83	11	4							
84	12	3							
85	3	1	0	-1					
86	1	2	0	-1					
87	2	4	0	-1					
88	4	3	0	-1					
89	3	2	0	-1					
90	4	1	0	-1					
91	7	11							
92	5	9							
93	6	10							
94	8	12							
95	7	9							
96	5	11							
97	5	10							
98	6	9							
99	6	12							
100	8	10							
101	7	12							
102	8	11							
103	11	9	0	+1					
104	9	10	0	+1					
105	10	12	0	+1					
106	12	11	0	+1					
107	11	10	0	+1					
108	12	9	0	+1					
16	3	3	33	48					
49	4	33	34	35	36				
50	4	37	38	39	40				
51	8	41	42	43	44	45	46	47	48

```
0.0200     0.0500     1.0000    40.2000    40.2000    0.0980    0.3000  10400.00
0.0200     0.0500     1.0000    40.2000    40.2000    0.0980    0.3000  10400.00
0.0200     0.0500     1.0000    40.2000    40.2000    0.0980    0.3000  10400.00
33    5   11    7     0
34    5   11    9
35    9    3   11
36    9    3    1
37   10    8    6
38   10    8   12
39   12    2    4
40   12    2   10
41   10    5    6
42   10    5    9
43    2    9   10
44    2    9    1
45    8   11    7
46    8   11   12
47   12    3   11
48   12    3    4
1.0000     1.0        1.0        1.0        1.0000
1
6
3    7    10   14    15   17
 1.0        1.0        1.0        1.0        1.0        1.0
4
7    8    11   12
 1.000      1.000      1.000      1.000
1
3
57   62   63
 1.0        1.0        1.0
2
21   25
 1.000      1.000
0
0
0
1.0000     1.0        1.0        1.0        1.0000
1
9
19   24   30   31    34   35    1    5    11
 1.0        1.0        1.0        1.0        1.0        1.0        1.0        1.0
 1.0
6
1    2    3    13    14   15
 1.000      1.000      1.000      1.000      1.000      1.000
0
0
0
0
0
0
1.0000     1.0        1.0        1.0        1.0000
0
0
0
1
6
40   45   47   51    52   54
 1.0        1.0        1.0        1.0        1.0        1.0
2
```

```
23    32
  1.000      1.000
1
3
94  100  102
  1.0        1.0        1.0
4
37    38   45    46
  1.000      1.000      1.000      1.000
1.0000     1.0        1.0        1.0        1.0000
0
0
0
0
0
0
1
9
91    96   102  103   106   107    73    77    83
  1.0        1.0        1.0        1.0        1.0        1.0        1.0        1.0
  1.0
6
33    34    35    45    46    47
  1.000      1.000      1.000      1.000      1.000      1.000
1.0000     1.0        1.0        1.0        1.0000
0
0
0
1
6
38    41    44    49    50    54
  1.0        1.0        1.0        1.0        1.0        1.0
2
20    28
  1.000      1.000
1
3
92    96    97
  1.0        1.0        1.0
4
33    34    41    42
  1.000      1.000      1.000      1.000
1.0000     1.0        1.0        1.0        1.0000
1
9
21    25    28    32    33    35     3     8     9
  1.0        1.0        1.0        1.0        1.0        1.0        1.0        1.0
  1.0
6
5     6     7     9     1    11
  1.000      1.000      1.000      1.000      1.000      1.000
0
0
0
0
0
0
0.086110   0.139120   0.041500   0.108700   0.041500   0.041500   0.041500   0.167490
0.296770   0.235440   0.067314   0.041500   0.041500   0.221420
  1    1    1    1    1    1    1    1    1    1    1    1    1    1
0.039328   0.050000   0.040024
```

```
   1      1      1
0.043365  0.046153  0.041500  0.041500  0.041500  0.041500  0.041500  0.049120
0.057969  0.041500  0.041500  0.041500  0.041500  0.060580
   1      1      1      1      1      1      1      1      1      1      1      1      1      1
0.050000  0.050000  0.037808
   1      1      1
0.041500  0.041500  0.041500  0.053185  0.041500  0.041500  0.189440  0.054885
0.073566  0.182260  0.122810  0.041500  0.041500  0.085213
   1      1      1      1      1      1      1      1      1      1      1      1      1      1
0.050000  0.041122  0.035357
   1      1      1
/*
//
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

D.2. Listing of the Program DIMCO

```
//FSSOS JOB (--------,30,30,2001),'DL NTDUC',TIME=25                    JOB  49
/*MESSAGEL     PLEASE INTERPRETE MY OUTPUT PUNCHED CARDS
// EXEC FORTCLG,REGION=150K,TIME=25
//FORT.SYSIN DD *
      INTEGER BNC,SN,CONS1,CONS2,CONS3,CONS4,CONS5,CONS9,CONS8,PN,SNN
      DIMENSION NJ(10),NBJ(10),NCB(10),NIC(10),NBW1(10),NBW2(10),NBW3(10
     1),NILJ(10)
      DIMENSION NM(50),NG(50),NW(50),MEB(50),MEF(50)
      DIMENSION ITY(3)
C
C     INPUT SOME CONTROL INFORMATION FOR ALL SUBSTRUCTURES
C
      READ(5,32)NN,NSU,NDAM,NLC,NV,NCC,BNC,NBW,NPH,NSD,ITE,NBLJ,NDMT    ,
     1LINK,ILIM
      READ(5,32)ITY(1),ITY(2),ITY(3)
C
C     CONTROL INFORMATIONS FOR EACH SUBSTRUCTURE
C
      IET=3
      KK=0
      DO 30 K=1,NSU
      READ(5,32)NJ(K),NBJ(K),NCB(K),NIC(K),NBW1(K),NBW2(K),NBW3(K),NILJ(
     2K)
 32   FORMAT(16I5)
      DO 31 J=1,IET
      IF(ITY(J).EQ.0)GO TO 31
      KK=KK+1
      READ(5,32)NM(KK),NG(KK),NW(KK),MEB(KK),MEF(KK)
 31   CONTINUE
 30   CONTINUE
C
C     INITIALIZED SOME VARIABLES
C
      NGU=-999
      KK=0
      CONS1=1
      CONS2=2
      CONS3=3
      CONS4=4
      CONS5=5
      CONS9=9
      CONS8=8
C
C     CALCULATION OF ALL SUBSCRIPTS USED IN DIMENSION STATEMENTS
C
      DO 1 I=1,NSU
      DO 1 J=1,IET
      IF(ITY(J).EQ.0)GO TO 1
      KK=KK+1
      IF(NG(KK).GT.NGU)NGU=NG(KK)
 1    CONTINUE
```

```
      KKU=NSU*ITE
      NBJL=-999
      NCIL=-999
      NCBL=-999
      NLJ=NBLJ
      NU3=0
      NU33=0
      NGG=0
      NTE=0
      NCE=0
      NSE=0
      M8=-999
      K3DUP=0
      NJKK=0
      NCII=0
      KK=0
      DO 2 I=1,NSU
      IF(NBJ(I).GT.NBJL)NBJL=NBJ(I)
      IF(NIC(I).GT.NCIL)NCIL=NIC(I)
      IF(NCB(I).GT.NCBL)NCBL=NCB(I)
      IF(NILJ (I).GT.NLJ)NLJ=NILJ(I)
      NU3=NU3+NBW3(I)
      IF(NBW3(I).GT.NU33)NU33=NBW3(I)
      DO 1000 J=1,IET
      IF(ITY(J).EQ.0)GO TO 1000
      KK=KK+1
      NGG=NGG+NG(I)
      GO TO(1001,1002,1003),J
1001  NTE=NTE+NM(KK)
      IF(NTE.GT.M8)M8=NTE
      GO TO 1000
1002  NCE=NCE+NM(KK)
      IF(NCE.GT.M8)M8=NCE
      GO TO 1000
1003  NSE=NSE+NM(KK)
      IF(NSE.GT.M8)M8=NSE
1000  CONTINUE
      K3DUP=K3DUP+NJ(I)
      NJKK=NJKK+NJ(I)
      IF(NIC(I).GT.NCII)NCII=NIC(I)
2     CONTINUE
      NMT=NTE+NCE+NSE
      K3EX=NTE
      IF(NCE.GT.K3EX)K3EX=NCE
      IF(NSE.GT.K3EX)K3EX=NSE
      SN=2*NN
      I108=2*SN
      PN=2*SN
      K1=NCIL
      IF(NV.GT.K1)K1=NV
      K2=NU3
      IF(NSD.GT.K2)K2=NSD
      K3DUP=K3DUP*SN
      K3=K3DUP
      IF(NPH.GT.K3)K3=NPH
      IF(K3EX.GT.K3)K3=K3EX
      K4=NMT
      IF(NPH.GT.K4)K4=NPH
      K5=BNC
      IF(NCIL.GT.K5)K5=NCIL
      K6=NBW
```

```
      IF(NU3.GT.K6)K6=NU3
      K66=NBW
      IF(NU33.GT.K66 )K66=NU33
      K7=NCBL+NLC
      IF(NSD.GT.K7)K7=NSD
      K9=NCBL
      IF(NSD.GT.K9)K9=NSD
      K10=NCIL
      IF(NSD.GT.K10)K10=NSD
      K11=NV
      IF(NPH.GT.K11)K11=NPH
      IF(NCC.GT.K11)K11=NCC
      LEN=-999
      I101=0
      DO 11 KA=1,NSU
      DO 11 I=1,ITE
      I101=I101+1
      IF(MEF(I101).GT.LEN)LEN=MEF(I101)
   11 CONTINUE
      K118=LEN
      IF(K11.GT.K118)K118=K11
      K12=3
      IF(NLC.GT.K12)K12=NLC
      K13=NMT
      IF(NPH.GT.K13)K13=NPH
      K20=NMT
      IF(NCC.GT.K20)K20=NCC
      IF(NSD.GT.K20)K20=NSD
      K21=ITE
      K22=NDMT
      K26=M8
      IF(NV.GT.K26)K26=NV
      I102=3*NTE+27*NCE+12*NSE
      I103=NTE+NCE+21*NSE
      I104=6*NTE+45*NCE+21*NSE
      I105=6*NTE+9*NCE+6*NSE
      I106=NTE
      IF(NJKK.GT.I106)I106=NJKK
      I107=NCII
      NUS=NSU
      IF(CONS2.GT.NUS)NUS=CONS2
      IPDAM=NDAM+1
      IETC=NV*IPDAM
C
C     TO AVOID SUBSCRIPT EQUAL ZERO
C
      I100=NDAM
      IF(NDAM.EQ.0)I100=NDAM+1
      IF(LINK.EQ.0)LINK=1
      IF(NCII.EQ.0)NCII=1
      IF(K22.EQ.0)K22=1
      IF(NCIL.EQ.0)NCIL=1
      IF(NU3.EQ.0)NU3=1
      SNN=SN
      IF(IPDAM.GT.SNN)SNN=IPDAM
      IF(NTE.EQ.0)NTE=1
      IF(NCE.EQ.0)NCE=1
      IF(NSE.EQ.0)NSE=1
C
C     BEGIN TO PUNCH ALL DIMENSION STATEMENTS ON CARDS
C
```

```
      WRITE(6,86)
      WRITE(6,81)
      WRITE(6,82)
      WRITE(6,40)BNC,NLC,NGU,KKU,ILIM,CONS2,NV,IET
      WRITE(6,41)CONS1,ILIM,I108,IET,NBJL,NSU,LINK,CONS2,NLJ,IET
      WRITE(6,42)CONS2,IPDAM
      WRITE(6,83)
      WRITE(6,84)
      WRITE(6,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
      WRITE(6,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
      WRITE(6,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
     1S9
      WRITE(6,46)CONS1,CONS9,CONS9
      WRITE(6,47)NGU,KKU,M8,K21,NGU,KKU,I102
      WRITE(6,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
      WRITE(6,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
      WRITE(6,50)CONS3,M8,K21
      WRITE(6,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
      WRITE(6,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
      WRITE(6,53)NSD,CONS8,NGU,KKU,NSD,NCC
      WRITE(6,54)CONS1,CONS1,CONS1,CONS1,CONS1,CONS1
      WRITE(6,55)NGU,KKU,BNC
      WRITE(6,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(6,57)CONS1,KKU
      WRITE(6,58)NSD,NSU,NSD,NSU,NLC
      WRITE(6,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
      WRITE(6,60) CONS1,NV
      WRITE(6,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
      WRITE(6,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(6,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(6,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(6,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
      WRITE(6,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(6,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(6,68)NGG,NSD
      WRITE(6,69)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
      WRITE(6,70) CONS2,CONS2,CONS2,CONS2,CONS2,CONS2
      WRITE(6,171)IETC,IPDAM,IPDAM
      WRITE(6,85)
      WRITE(7,86)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,40)BNC,NLC,NGU,KKU,ILIM,CONS2,NV,IET
      WRITE(7,41)CONS1,ILIM,I108,IET,NBJL,NSU,LINK,CONS2,NLJ,IET
      WRITE(7,42)CONS2,IPDAM
      WRITE(7,83)
      WRITE(7,84)
      WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
      WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
      WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
     1S9
      WRITE(7,46)CONS1,CONS9,CONS9
      WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
      WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
      WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
      WRITE(7,50)CONS3,M8,K21
      WRITE(7,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
      WRITE(7,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
      WRITE(7,53)NSD,CONS8,NGU,KKU,NSD,NCC
      WRITE(7,54)CONS1,CONS1,CONS1,CONS1,CONS1,CONS1
      WRITE(7,55)NGU,KKU,BNC
```

```
      WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(7,57)CONS1,KKU
      WRITE(7,58)NSD,NSU,NSD,NSU,NLC
      WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
      WRITE(7,60) CONS1,NV
      WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
      WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
      WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(7,68)NGG,NSD
      WRITE(7,69)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
      WRITE(7,70) CONS2,CONS2,CONS2,CONS2,CONS2,CONS2
      WRITE(7,171)IETC,IPDAM,IPDAM
      WRITE(7,85)
      WRITE(7,87)
      WRITE(7,84)
      WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
      WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
      WRITE(7,85)
      WRITE(7,88)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,83)
      WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
     1S9
      WRITE(7,46)CONS1,CONS9,CONS9
      WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
      WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
      WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
      WRITE(7,50)CONS3,M8,K21
      WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(7,57)CONS1,KKU
      WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(7,85)
      WRITE(7,89)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,83)
      WRITE(7,84)
      WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
      WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
      WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
     1S9
      WRITE(7,46)CONS1,CONS9,CONS9
      WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
      WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
      WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
      WRITE(7,50)CONS3,M8,K21
      WRITE(7,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
      WRITE(7,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
      WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(7,57)CONS1,KKU
      WRITE(7,58)NSD,NSU,NSD,NSU,NLC
      WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
```

```
 WRITE(7,60) CONS1,NV
 WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
 WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
 WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
 WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
 WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
 WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
 WRITE(7,71)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
 WRITE(7,70) CONS2,CONS2,CONS2,CONS2,CONS2,CONS2
 WRITE(7,85)
 WRITE(7,90)
 WRITE(7,81)
 WRITE(7,82)
 WRITE(7,83)
 WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
1S9
 WRITE(7,46)CONS1,CONS9,CONS9
 WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
 WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
 WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
 WRITE(7,50)CONS3,M8,K21
 WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
 WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
 WRITE(7,85)
 WRITE(7,91)
 WRITE(7,81)
 WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
 WRITE(7,85)
 WRITE(7,92)
 WRITE(7,81)
 WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
 WRITE(7,85)
 WRITE(7,94)
 WRITE(7,81)
 WRITE(7,82)
 WRITE(7,83)
 WRITE(7,84)
 WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
 WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
 WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
1S9
 WRITE(7,46)CONS1,CONS9,CONS9
 WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
 WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
 WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
 WRITE(7,50)CONS3,M8,K21
 WRITE(7,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
 WRITE(7,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
 WRITE(7,54)CONS1,CONS1,CONS1,CONS1,CONS1,CONS1
 WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
 WRITE(7,60) CONS1,NV
 WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
 WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
 WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
 WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
 WRITE(7,69)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
 WRITE(7,85)
 WRITE(7,90)
 WRITE(7,81)
 WRITE(7,82)
 WRITE(7,83)
```

```
 WRITE(7,84)
 WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
 WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
 WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
1S9
 WRITE(7,46)CONS1,CONS9,CONS9
 WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
 WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
 WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
 WRITE(7,50)CONS3,M8,K21
 WRITE(7,54)CONS1,CONS1,CONS1,CONS1,CONS1,CONS1
 WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
 WRITE(7,57)CONS1,KKU
 WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
 WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
 WRITE(7,69)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
 WRITE(7,85)
 WRITE(7,96)
 WRITE(7,81)
 WRITE(7,82)
 WRITE(7,83)
 WRITE(7,84)
 WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
 WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
 WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
1S9
 WRITE(7,46)CONS1,CONS9,CONS9
 WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
 WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
 WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
 WRITE(7,50)CONS3,M8,K21
 WRITE(7,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
 WRITE(7,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
 WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
 WRITE(7,57)CONS1,KKU
 WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
 WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
 WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
 WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
 WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
 WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
 WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
 WRITE(7,85)
 WRITE(7,97)
 WRITE(7,81)
 WRITE(7,82)
 WRITE(7,83)
 WRITE(7,84)
 WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
 WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
 WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
1S9
 WRITE(7,46)CONS1,CONS9,CONS9
 WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
 WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
 WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
 WRITE(7,50)CONS3,M8,K21
 WRITE(7,51)CONS1,CONS1,IPDAM,IPDAM,IPDAM,IPDAM,IPDAM
 WRITE(7,52)CONS1,K22,IPDAM,K22,KKU,I100,NSU,IPDAM
 WRITE(7,53)NSD,CONS8,NGU,KKU,NSD,NCC
 WRITE(7,55)NGU,KKU,BNC
```

```
      WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(7,57)CONS1,KKU
      WRITE(7,58)NSD,NSU,NSD,NSU,NLC
      WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
      WRITE(7,60) CONS1,NV
      WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
      WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
      WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(7,85)
      WRITE(7,98)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,83)
      WRITE(7,84)
      WRITE(7,53)NSD,CONS8,NGU,KKU,NSD,NCC
      WRITE(7,58)NSD,NSU,NSD,NSU,NLC
      WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
      WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
      WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(7,85)
      WRITE(7,99)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,83)
      WRITE(7,84)
      WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
      WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
      WRITE(7,45)CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,CONS9,K20,CON
     1S9
      WRITE(7,46)CONS1,CONS9,CONS9
      WRITE(7,47)NGU,KKU,M8,K21,NGU,KKU,I102
      WRITE(7,48)CONS1,I103,I104,IET,M8,K21,M8,K21,M8
      WRITE(7,49)CONS2,K21,NGU,KKU,M8,K21,I105,IET,IET
      WRITE(7,50)CONS3,M8,K21
      WRITE(7,53)NSD,CONS8,NGU,KKU,NSD,NCC
      WRITE(7,56)NCIL,NLC,NSU,M8,K21,NGU,KKU,M8,KKU,NGU
      WRITE(7,57)CONS1,KKU
      WRITE(7,58)NSD,NSU,NSD,NSU,NLC
      WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
      WRITE(7,60) CONS1,NV
      WRITE(7,61)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
      WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
      WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
      WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
      WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
      WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
      WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
      WRITE(7,68)NGG,NSD
      WRITE(7,85)
      WRITE(7,300)
      WRITE(7,81)
      WRITE(7,82)
      WRITE(7,83)
```

```
WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
WRITE(7,60) CONS1,NV
WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
WRITE(7,85)
WRITE(7,301)
WRITE(7,81)
WRITE(7,82)
WRITE(7,83)
WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
WRITE(7,85)
WRITE(7,302)
WRITE(7,81)
WRITE(7,82)
WRITE(7,83)
WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
WRITE(7,60) CONS1,NV
WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
WRITE(7,64)I106,NSU,NPH,NPH,K4,NV,NV
WRITE(7,65)K5,K66,K5,K7,BNC,K9,NCII,NU3,NSU
WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
WRITE(7,85)
WRITE(7,303)
WRITE(7,81)
WRITE(7,62)M8,K21,NTE,NLC,NCE,NLC,CONS4,NSE,NLC,CONS3
WRITE(7,63)CONS1,NV,NUS,NPH,M8,K21,K3
WRITE(7,66)K10,K9,K11,K12,K118,K12,K11,K26
WRITE(7,67)CONS1,K13,M8,NSU,NCBL,NSU
WRITE(7,85)
WRITE(7,304)
WRITE(7,81)
WRITE(7,82)
WRITE(7,83)
WRITE(7,84)
WRITE(7,43)NSU,KKU,KKU,NSU,NSU,NSU,KKU
WRITE(7,44)CONS1,NSU,NSU,NSU,NSU,NSU,KKU,KKU
WRITE(7,59)NGU,KKU,NGU,KKU,NGU,KKU,K1,K2,NCIL,NSU
WRITE(7,60)CONS1,NV
WRITE(7,611)NCIL,NCBL,NSU,NCIL,NLC,NSU,BNC,NBW,BNC,NLC
WRITE(7,72)K5,K66,K5,K7,BNC,K9,NCII,NU3
WRITE(7,73)CONS1,NSU
WRITE(7,666)K10,K9,K11,K12,K118,K12,K11,K26
WRITE(7,677)CONS1,K13,M8,NSU,NCBL,NSU
WRITE(7,74)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1,SNN
WRITE(7,75)CONS2,CONS2,CONS2,CONS2,CONS2,CONS2
WRITE(7,85)
WRITE(7,306)
WRITE(7,81)
WRITE(7,76)K5,K66,K5,K7,BNC,K9,NCII,NU3
WRITE(7,77)CONS1,NSU
WRITE(7,78)NCC,CONS2,NCC,CONS2,CONS2,CONS1,CONS1, SNN
WRITE(7,79)CONS2,CONS2,CONS2,CONS2,CONS2,CONS2
```

```
      WRITE(7,85)
C
C     FORMAT PUNCHING STATEMENTS
C
81    FORMAT(6X,'IMPLICIT REAL*8 (A-H,O-Z)')
82    FORMAT(6X,'INTEGER SIZE,BNC,SN')
40    FORMAT(6X,'DIMENSION PB(',I3,',',I2,'),ALP(',I3,',',I2,'),DBIN(',I
     13,',',I2,'),OO(',I3,'),FACC(',I2,'),FB(')
41    FORMAT(5X,I1,I3,'),BETA(',I2,'),CL(',I2,'),NZ(',I3,',',I2,'),LINLG
     1(',I2,',',I2,'),NJL(',I2,'),NVV(',I2,'),NEGV(')
 42   FORMAT(5X,I1,I2,')')
83    FORMAT(6X,'COMMON STEP,BNC,SN,NBW,SIZE,NLC,NSU')
84    FORMAT(6X,'COMMON/V1/N1,NCI,NWK,NGK,MA,NU1,NU2,NU3,M1,NB,NJK,NC,N1
     21,ISQ,IQ1')
43    FORMAT(6X,'COMMON/V2/NIC(',I2,'),NW(',I2,'),NG(',I2,'),NBW1(',I2,'
     1),NBW2(',I2,'),NBW3(',I2,'),NM(',I2,'),')
44    FORMAT(5X,I1,'NBJ(',I2,'),NJ(',I2,'),NCB(',I2,'),NEW(',I2,'),IQS('
     1,I2,'),MEB(',I2,'),MEF(',I2,')')
 45   FORMAT(6X,'COMMON/P1/B1(',I2,',',I2,'),B2(',I2,',',I2,'),B3(',I2,'
     1,',I2,'),ESF(',I2,',',I2,'),NA(',I4,'),NI1(',I2)
46    FORMAT(5X,I1,'),NJ1(',I2,'),NJ2(',I2,')')
47    FORMAT(6X,'COMMON/P2/XNUU(',I3,',',I2,'),ELL(',I3,',',I2,'),BU(',I
     13,',',I2,'),STRESS(',I4,'),TCSM(')
48    FORMAT(5X,I1,I4,'),TRCSSP(',I4,'),XCOST(',I2,'),ICSS(',I4,',',I2,'
     1),ISAC(',I4,',',I2,'),INDC(',I4)
49    FORMAT(5X,I1,',',I2,'),IGRT(',I3,',',I2,'),IGRE(',I4,',',I2,'),NND
     1C(',I5,'),LLN(',I2,'),ITY(',I2,'),ICSSM(')
50    FORMAT(5X,I1,I4,',',I2,')')
51    FORMAT(6X,'COMMON/P3/EVEC(',I3,',',I2,'),RRF(',I2,'),RDLIM(',I2,')
     1,RSL(',I2,'),RSU(',I2,'),RLOAD(',I2,')')
52    FORMAT(5X,I1,',REDUC(',I3,'),NDOF(',I2,'),NDM(',I3,'),NBDAM(',I2,'
     1,',I2,'),KIIDAM(',I2,',',I2,')')
53    FORMAT(6X,'COMMON/P4/INF(',I3,',',I2,'),NGV(',I3,',',I2,'),INO(',I
     13,'),NDISP(',I3,')')
54    FORMAT(6X,'COMMON/P5/YK(',I3,'),YM(',I3,'),SK(',I3,'),SM(',I3,'),E
     1Y(',I3,'),SG(',I3,')')
55    FORMAT(6X,'COMMON/R1/BL(',I3,',',I2,'),DLIB(',I3,')')
56    FORMAT(6X,'COMMON/R2/PI(',I2,',',I2,',',I2,'),RR(',I4,',',I2,'),E(
     1',I3,',',I2,'),MN(',I4,',',I2,'),NOM(',I3,',')
57    FORMAT(5X,I1,I2,')')
58    FORMAT(6X,'COMMON/R4/IIL(',I3,',',I2,'),KLC(',I3,'),IOK(',I2,'),NO
     1(',I2,')')
59    FORMAT(6X,'COMMON/R5/B(',I3,',',I2,'),SL(',I3,',',I2,'),SU(',I3,',
     1',I2,'),DPB(',I3,',',I3,'),DLIM(',I2,',',I2,')')
60    FORMAT(5X,I1,',SS(',I3,')')
61    FORMAT(6X,'COMMON/A1/Q(',I2,',',I3,',',I2,'),ZI(',I2,',',I2,',',I2
     1,'),C(',I3,',',I3,'),ZB(',I3,',',I2,')')
611   FORMAT(6X,'COMMON/A1/G(',I2,',',I3,',',I2,'),ZI(',I2,',',I2,',',I2
     1,'),C(',I3,',',I3,'),ZB(',I3,',',I2,')')
62    FORMAT(6X,'COMMON/A3/BR(',I4,',',I2,'),TRSF(',I4,',',I2,'),CSTF(',
     1I2,',',I2,',',I2,'),SSPF(',I2,',',I2,',',I2,'),')
63    FORMAT(5X,I1,'Z(',I3,',',I2,'),DZE(',I3,'),MP(',I4,',',I2,'),ND(',
     1I5,')')
64    FORMAT(6X,'COMMON/A4/X(',I4,',',I2,'),DLP(',I3,'),DLPH(',I3,'),T('
     1,I4,'),WM(',I3,'),RO(',I3,')')
65    FORMAT(6X,'COMMON/A5/D(',I3,',',I3,'),DS(',I3,',',I3,'),A2(',I3,',
     1',I3,'),DKI(',I2,',',I2,'),KIIUBW(',I2,')')
66    FORMAT(6X,'COMMON/A6/DPZ(',I3,',',I3,'),ZZ(',I3,',',I2,'),BE(',I4,
     1',',I2,'),W(',I3,'),H(',I4,'),VV(')
666   FORMAT(6X,'COMMON/A6/DPZ(',I3,',',I3,'),ZZ(',I3,',',I2,'),BE(',I4,
     1',',I2,'),F(',I3,'),H(',I4,'),VV(')
```

```
67    FORMAT(5X,I1,I4,'),Y(',I4,',',I2,'),NZC(',I3,',',I2,')')
677   FORMAT(5X,I1,I4,'),A(',I4,',',I2,'),NZC(',I3,',',I2,')')
68    FORMAT(6X,'COMMON/A7/DPX(',I3,',',I3,')')
69    FORMAT(6X,'COMMON/C1/XEIG(',I3,',',I2,'),YXEIG(',I3,',',I2,'),WS('
     1,I2,'),DM(',I3,',',I3,'),IET(',I2,')')
70    FORMAT(5X,'$/C3/ QQK(',I2,',',I2,'),QQM(',I2,',',I2,'),QA(',I2,','
     1,I2,')')
71    FORMAT(6X,'COMMON/C1/XEIG(',I3,',',I2,'),YXEIG(',I3,',',I2,'),WS('
     1,I2,'),DM(',I3,',',I3,'),IETA(',I2,')')
72    FORMAT(6X,'COMMON/A5/XCL(',I3,',',I3,'),AMAS2(',I3,',',I3,'),XM(',
     1I3,',',I3,'),DKI(',I2,',',I2,'),KIIUB')
73    FORMAT(5X,I1,'W(',I2,')')
74    FORMAT(6X,'COMMON/C1/X(',I3,',',I2,'),Y(',I3,',',I2,'),W(',I2,'),D
     1M(',I3,',',I3,'),IETA(',I2,')')
75    FORMAT(5X,'$/C3/ QQK(',I2,',',I2,'),QQM(',I2,',',I2,'),Q(',I2,',',
     1I2,')')
76    FORMAT(6X,'COMMON/A5/XCL(',I3,',',I3,'),AMAS2(',I3,',',I3,'),ZM(',
     1I3,',',I3,'),DKI(',I2,',',I2,'),KIIUB')
77    FORMAT(5X,I1,'W(',I2,')')
78    FORMAT(6X,'COMMON/C1/Z(',I3,',',I2,'),Y(',I3,',',I2,'),W(',I2,'),D
     1M(',I3,',',I3,'),IETA(',I2,')')
79    FORMAT(5X,'$/C3/ XK(',I2,',',I2,'),XM(',I2,',',I2,'),P(',I2,',',I2
     1,')')
171   FORMAT(6X,'COMMON/C4/ETC(',I4,'),TEI(',I2,'),TE(',I2,')')
85    FORMAT(6X,'**************************************************')
86    FORMAT(6X,'$$$$$        MAIN')
87    FORMAT(6X,'$$$$$        VARI')
88    FORMAT(6X,'$$$$$        ELESTF')
89    FORMAT(6X,'$$$$$        STIFFM')
90    FORMAT(6X,'$$$$$        RECALL')
91    FORMAT(6X,'$$$$$        DECUPP')
92    FORMAT(6X,'$$$$$        SOLDUP')
94    FORMAT(6X,'$$$$$        MKYS')
95    FORMAT(6X,'$$$$$        DEFREQ')
96    FORMAT(6X,'$$$$$        ZBZIEF')
97    FORMAT(6X,'$$$$$        CONST')
98    FORMAT(6X,'$$$$$        ABSMAX')
99    FORMAT(6X,'$$$$$        GENC')
300   FORMAT(6X,'$$$$$        DELBE')
301   FORMAT(6X,'$$$$$        DESVV')
302   FORMAT(6X,'$$$$$        SDD')
303   FORMAT(6X,'$$$$$        SOLVEL')
304   FORMAT(6X,'$$$$$        SUBSP')
306   FORMAT(6X,'$$$$$        JACOBI')
      IDIM=BNC*NLC+NGU*KKU+ILIM*CONS2+NV+IET+ILIM+I108+IET+NBJL*NSU+LINK
     1*CONS2+NLJ+IET+IPDAM
      IV2=9*NSU+5*KKU
      IP1=4*81+3*9+K20
      IP2=3*NGU*KKU+M8*K21+I102+I103+I104+IET+5*M8*K21+I105+2*IET
      IP3=1+6*IPDAM+2*K22+KKU*I100+NSU*IPDAM
      IP4=NSD*8+NGU*KKU+NSD+NCC
      IP5=6
      IR1=NGU*KKU+BNC
      IR2=NCIL*NLC*NSU+M8*K21*1+2*NGU*KKU+M8*KKU
      IR4=NSD*NSU+NSD+NSU+NLC
      IR5=3*NGU*KKU+K1*K2+NCIL*NSU+NV
      IA1=NCIL*NCBL*NSU+NCIL*NLC*NSU+BNC*NBW+BNC*NLC
      IA3=2*M8*K21+NTE*NLC*1+4*NCE*NLC+NSE*NLC*3+NV*NUS+NPH+K3
      IA4=I106*NSU+2*NPH+K4+2*NV
      IA5=K5*K66+K5*K7+BNC*K9+NCII*NU3+NSU
      IA6=K10*K9+K11*K12+K118*K12+K11+K26+K13+M8*NSU+NCBL*NSU
```

```
      IA7=NGG*NSD
      IC1=2*NCC*2+3+SNN
      IC3=12
      IC4=IETC+2*IPDAM
      MEMO=IDIM+IV2+IP1+IP2+IP3+IP4+IP5+IR1+IR2+IR4+IR5+IA1+IA3+IA4+IA5+
     1IA6+IA7+IC1+IC3 +IC4
      WRITE(6,33)MEMO
 33   FORMAT(6X,'TOTAL MEMORIES USED = ',I6)
      WRITE(6,200)
 200  FORMAT(10X,'SUCESSFUL RUN')
      STOP
      END
//GO.SYSIN DD
```

LIST OF SYMBOLS

B	a subscript used to indicate quantities assiciated with boundary coordinates
b	a vector of design variables
b^L	lower bound on b
b^U	upper bound on b
$C^{(\alpha)}$	a matrix defined in Equation 2.4-11
$C_1^{(\alpha)}$	a matrix defined in Equation 2.4-12
$C_2^{(\alpha)}$	a matrix defined in Equation 2.4-4
D	total number of design variables
d	superscript for design variabel constraint
$\overline{d}$	total number of damage condition
e	superscript for eigenvalue constraint
F_B	a vector of effective boundary forces for the entire structure
f	natural frequency (Hz)
$G^{(\alpha)}$	a matrix defined in Equation 2.4-18
H	a matrix defined in Equation 2.5-12
I	a subscript used to indicate quantities associated with interior coordinate
I_i	moment of inertia of the i^{th} member
J	cost function defined by Equation 2.3-12
K(b)	stiffness matrix for the entire structure; (N x N)
K_B	boundary stiffness matrix for the entire structure; (n x n)
K_{BB}, K_{BI}, K_{IB}, K_{II}	submatrices of K(b)
L	total number of substructures
ℓ_{ij}	length or surface area of the j^{th} member in the i^{th} group
ℓ_i	equivalent length of the i^{th} member
M(b)	mass matrix for the entire structure; (N x N)
m	total number of interior degrees of freedom
m(r)	number of interior degrees of freedom for the r^{th} substructure
N	total number of degrees of freedom

n	total number of boundary degrees of freedom
$n(r)$	number of boundary degrees of freedom for the r^{th} substructure
$p^{(r)}$	a vector of member forces for the r^{th} substructure
r	superscript for r^{th} substructure
$\overline{r}$	cost function reduction ratio, needed in calculating the step size
$S(b)$	a vector of externally applied loads
S_B	a subvector of S associated with the boundary degrees of freedom
S_I	a subvector of S associated with the interior degrees of freedom
s	superscript for superscript for stress and displacement constraints
W	weighting matrix
W_i	coefficient of weighting matrix associated with i^{th} design variable
$\overline{w}_i$	multiplier associated with W
x_1, x_2, x_3	cartesian coordinates
$y^{(\alpha)}$	eigenvector associated with Equation 2.2-16
$z^{(\alpha)}$	a vector of nodal displacements for the entire structure
z^a	a vector of allowable nodal displacements for the entire structure
$z_B^{(\alpha)}$	a vector of boundary displacements for the entire structure
$z_I^{(\alpha)}$	a vector of interior displacements for the entire structure
δb	a vector of small changes of design variable b
$\delta z_I^{(\alpha)}$	a vector of small changes in the vector z_I
$\delta z_B^{(\alpha)}$	a vector of small changes in the vector z_B
δb^1, δb^2	defined in Equations 2.5-9 and 2.5-10
α	a superscript used to denote a damage condition
$\overline{\alpha}_i$	positive constant used to calculate the moment of inertia
$\beta(r)$	a Boolean transformation matrix for the r^{th} substructure
σ^a	an allowable stress

σ^c	calculated stress
ρ_i	material density of members of the i^{th} group
μ	Lagrange multiplier vector
μ^1, μ^2	components of μ
η	step size used in Equation 2.5-8
ζ	eigenvalues associated with Equation 2.2-16
$\phi^{s(\alpha)}, \phi^d$	vector constraint functions used in Equation 2.5-6
ϕ^e	scalar frequency constraint function used in Equation 2.5-6
λ_I, λ_B	adjoint matrices obtained from Equation 2.4-28
$\left.\begin{matrix}\lambda_I^{J(\alpha)}, \lambda_B^{J(\alpha)} \\ \lambda_B^{s(\alpha)}, \lambda_I^{s(\alpha)}\end{matrix}\right\}$	adjoint matrices obtained from Equations 2.4-28, 2.4-29, 2.4-24 and 2.4-25
$\left.\begin{matrix}\Lambda^J, \Lambda^{s(\alpha)} \\ \Lambda^d, \Lambda^{e(\alpha)}\end{matrix}\right\}$	matrices whose columns represent sensitivity vectors defined in Equations 2.4-27, 2.4-33, 2.4-21 and 2.4-19.

<u>ABBREVIATIONS</u>

CST	constant strain triangular elements
NLC	number ofloading conditions
FSODPS	fail-safe design problem with substructuring
SSP	symmetric shear panel
SPSP	symmetric pure shear panel
TP(r)	number of element types in the r^{th} substructure
SOS	structural optimization with substructures
DOF	degrees of freedom
DIMCO	<u>Dim</u>ension <u>Co</u>mputer